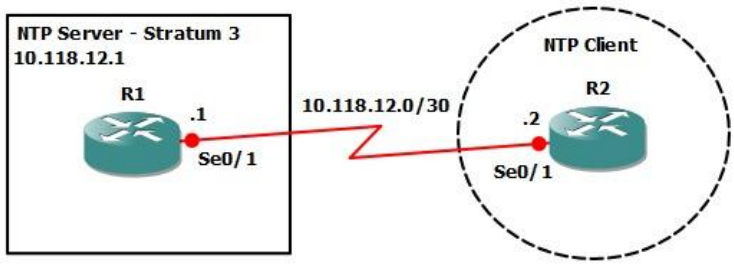
# Chapter 1: "Planning Maintenance for Complex Networks"

**Fundamental Tools, Applications, and Resources:**

**NTP Server Configuration Example:**

The following configuration tells the router to synchronize it's time and date with another NTP Server:




R1:

```
clock set 20:00:00 aug 26 2010
!
ntp master 3
```

R2:

```
ntp server 10.118.12.1
```

```
show ntp associations
```

the following commands tell the router to use time stamps for debug and log commands:

```
service time stamps debug datetime msec localtime show-timezone
```

```
service time stamps log datetime localtime show-timezone
```

Note:

```
ntp master 3
```

3 is the stratum "priority of the NTP server "The position of a device in the NTP hierarchy is determined by its stratum, which serves as an NTP hop count. A stratum 1 server is a server that is directly connected to an authoritative time source such as a radio or atomic clock. A server that synchronizes its clock to a stratum 1 server will become a stratum 2 time source, and so on.

**NOTE**: you can use external NTP severs for example: africa.pool.ntp.org, asia.pool.ntp.org

**Logging Services:**

Syslog has been illustrated before in the switching course " summary page 32":

By default, all messages from level 0 to 7 are logged to the console; however, similar to buffer logging, you can configure the severity level as an optional parameter on the **logging console** command.

Configuration Example:

To keep tracking logs even if the device is rebooted or shutdown, use external syslog server for example, Kiwisyslog is a good free one:

```
Logging 10.1.152.1
Logging console
Logging trap
```

Last 2 commands are the default.

This will send all logs to the kiwisyslog.

**Implementing Backup and Restore Services**

The fact that the TFTP protocol does not require any authentication and that all content is sent across the network in clear text makes it a relatively unsecure mechanism. More secure protocols such as FTP, SCP, and HTTP or HTTPS can also be used as a means of transferring configurations and software. To use any of these more-secure protocols, you must specify the username and password that are used to authenticate to the server. For all of these protocols, the credentials can be specified as part of the Uniform Resource Locator (URL) that is used with the copy command. The username and password are specified by placing the username and password as username:password@ before the server name or IP address in the URL.

**FTP:**

You can use the following command to copy files to and FTP server:

```
copy startup-config ftp://username:password@FTP-server/file-name
copy startup-config ftp://backup:san-fran@10.1.152.1/RO1-test.cfg
```

For SCP, HTTP and HTTPS you would use a similar syntax, replacing the URL prefix ftp:// with scp://, http:// or https://, respectively.

Otherwise you cand specify username and password in advance to avoid writing it again in the above command:

```
ip ftp username backup
ip ftp password san-fran
```

```
ip http client username backup
ip http client password 0 san-fran
copy startup-config ftp://10.1.152.1/RO1-test.cfg
```

It is important to know that even though FTP and HTTP require authentication, these protocols send credentials in clear text. HTTPS and SCP use encryption to ensure confidentiality of both the transmitted credentials and the content of the transferred file. When possible, use secure protocols such as HTTPS and SCP.

Setting Up the Configuration Archive:

The storage place can either be a local or a networked path supported by the Cisco IOS file system. Not all types of local flash storage are supported, so check your device's flash type for support of this feature if you want to store your configuration archive locally on a device instead of on a server. The configuration path can include the variables **$h** for the device's hostname and **$t** to include a time and date stamp in the filename.

the biggest advantage of this feature is the way you can use it to create and update a configuration archive automatically. By adding the **write-memory** option to the archive configuration section, you can trigger an archive copy of the running configuration to be created any time the running configuration is copied to NVRAM. It is also possible to generate archive copies of the configuration periodically by specifying the **time-period** option followed by a time period, specified in minutes. Each time the configured time period elapses, a copy of the running configuration will be archived.

```
archive
 path flash:/config-archive/$h-config$t or
 path ftp://192.168.10.1/$h-config$t
 write-memory
 time-period 10080
```

```
show archive
```

you can make a new folder  in flash:

```
mkdir flash: "name"
```

Note:

In GNS3 and CoreFTP it worked but after removing the $t, and it named files, 1, 2, 3, …

configure replace command:

The **configure replace** command enables you to replace the currently running configuration on the router with a saved configuration. It does so by comparing the running configuration with the configuration file appointed by the **configure replace** command and then creates a list of differences between the files. Based on the discovered differences, various Cisco IOS configuration commands are generated that will change the existing running configuration to the replacement configuration. The device does not need to be reloaded

```
TEST# configure replace flash:config-archive/RO1-config-5 list
```

The command option **list** is added to the **configure replace** command to show the configuration commands that are being applied by the configuration replacement.

Measure the CPU load:

```
show processes cpu
show processes cpu history
```
this last command will display a chart showing the history of CPU utilization.

**CDP:**

These are some useful show commands for CDP:

#show CDP neighbors

#show CDP neighbors brief

#show CDP entry "*hostname*"*** >> note that * is used with no spaces after the hostname.**

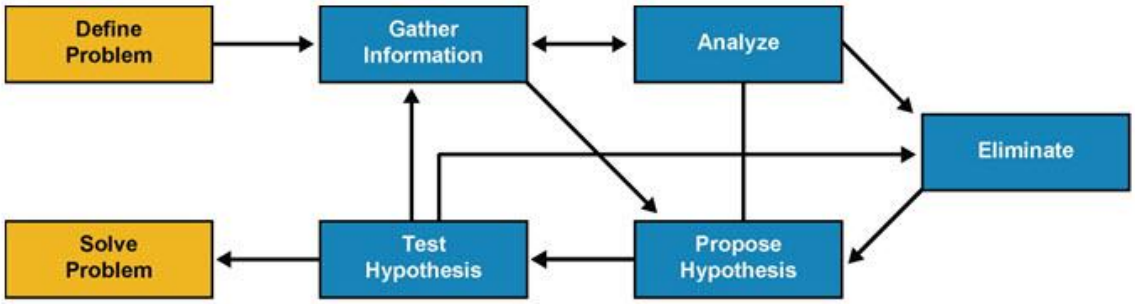
#show CDP interface >> this will show all interfaces that run CDP.

**Tracert:**

Tracert –d "ip address" >> this command is used on the **CMD**, where –d does not resolve the host name which takes longer.

# Chapter 2. Troubleshooting Processes for Complex Networks

Flow Chart of a Structured Troubleshooting Approach:




Step 1. Defining the problem

Step 2. Gathering information

Step 3. Analyzing the information

Step 4. Eliminating possible problem causes

Step 5. Formulating a hypothesis about the likely cause of the problem

Step 6. Testing that hypothesis

Step 7. Solving the problem

## Structured Troubleshooting Approaches:

Commonly used troubleshooting approaches include the following:

• **Top down:** Using this approach, you work from the Open Systems Interconnection (OSI) model's application layer down to the physical layer.

• **Bottom up:** The bottom-up approach starts from the OSI model's physical layer and moves up to the application layer.

• **Divide and conquer:** Using this approach, you start in the middle of the OSI model's stack (usually the network layer) and then, based on your findings, you move up or down the OSI stack.in this method you use the PING utility and then decide where to go next, up or down.

• **Follow the path:** This approach is based on the path that packets take through the network from source to destination.

• **Spot the differences:** As the name implies, this approach compares network devices or processes that are operating correctly to devices or processes that are not operating as expected and gathers clues by spotting significant differences. In case the problem occurred after a change on a single device was implemented, the spot-the-differences approach can pinpoint the problem cause by focusing on the difference between the device configurations, before and after the problem was reported.

• **Move the problem:** The strategy of this troubleshooting approach is to physically move components and observe whether the problem moves with the components.

# Chapter 3. Using Maintenance and Troubleshooting Tools and Applications

Collecting and Filtering Information Using Cisco IOS show Commands:

Filtering depends on Regular expressions which are patterns that can be used to match strings in a piece of text. In its simplest form, you can use it to match words or text fragments in a line of text, but full use of the regular expression syntax allow you to build complex expressions that match specific text patterns.

```
show ip route 10.1.193.3
```

Router will respond if it has a route to reach the past host "host not subnet".

```
show ip route 10.1.193.10
  % Subnet not in table
```

Keep in mind that if gateway of last resort (default route) is present in the IP routing table, but no entry matches the IP address you entered, the router again responds with the **% Subnet not in table** message even though packets for that destination are forwarded using the gateway of last resort.

```
show ip route 10.1.193.0 255.255.255.0 longer-prefixes
```

The router will then list all subnets that fall within the prefix that you have specified (including that prefix itself, if it is present in the routing table). If the network that you are troubleshooting has a good hierarchical IP numbering plan, the **longer-prefixes** command option can prove useful for displaying addresses from a particular part of the network. You can display all subnets of a particular branch office or data center, for example, using the summary address for these blocks and the **longer–prefixes** keyword.

In this example it will show all subnets that has /24 or less.

```
show processes cpu | include IP Input
  71   3149172  7922812  397  0.24%  0.15%  0.05% 0  IP Input
```

You can also exclude all processes that does not use any processing just to customize the results:

```
show processes cpu | exclude 0.00%
```

you are only interested in the IP Input process in the output of the **show processes cpu** command, so you select only the lines that contain the string "IP Input" by using the command **show processes cpu | include IP Input**. Note that IP Input is case sensitive.

```
show ip interface brief | exclude unassigned
```

this will display only interfaces that has an IP address.

```
show running-config | begin line vty
```

to begin with a certain line in the long running configs.

```
show processes cpu | include ^CPU|IP Input
```

the caret (^) character, which is used to denote that a particular string will be matched only if it occurs at the beginning of a line. The expression **^CPU** will therefore only match lines that start with the characters "CPU" and not any line that contains the string "CPU". The same line uses the pipe character (|) (without a preceding and following space) as part of a regular expression to signify a logical OR. As a result, the **show processes cpu | include ^CPU|IP Input** command displays only the lines that start with the string "CPU" or contain the string "IP Input".

Section Option:

Cisco IOS Software Release (12.3(2)T) introduced the **section** option, which allows you to select and display a specific section or lines from the configuration that match a particular regular expression and any following associated lines.

```
show running-config | section router eigrp
show access-lists | section standard
```

Using the redirect, append, and tee options with show Commands:

```
show tech-support | redirect tftp://192.168.37.2/show-tech.txt
```

! The **redirect** option does not display the output on the console

```
show ip interface brief | tee flash:show-int-brief.txt
```

! The **tee** option displays the output on the console and send it to the file

```
show version | append flash:show-commands.txt
show ip interface brief | append flash:show-commands.txt
```

! The append option allows you to add the command output to an existing file instead of replacing that file. The use of this command option makes it easy to collect the output of several **show**

commands in a text file. A prerequisite for this option is that the file system that you are writing to must support “append” operations; so for instance, a TFTP server cannot be used in this case.

Note:

The show tech-support command collects information about your devices when you need to send it to for example cisco support when you have a problem.

Testing Network Connectivity Using ping and Telnet

ping utility:

The ping utility has some extended options that are useful for testing specific conditions, including the following:

**repeat-count:** The repeat option enables you to send out hundreds to thousands of packets to help pinpoint a pattern in the occurrence of the packet loss. The default are 5 packets.

**datagram-size:** the size option allows you to determine the maximum transmission unit (MTU) along the path to a particular destination IP address. Note that in windows the default packet size is 32 bytes, but in cisco the default packet size is 100-byte.

**source [address | interface]:** This option allows you to set the source IP address or interface of the ping packet. The IP address has to be one of the local device’s own IP addresses. If this option is not used, the router will select the IP address of the egress interface as the source of the ping packets. If the ping fails when you use a different source interface the reason may be one of the routers on the return path does not have a route to the address/subnet of the source interface.

There might be several other reasons for this, too. For example, an access list on one of the transit routers might be blocking the IP address of the source interface. Specifying the source IP address or interface proves useful when you want to check two-way reachability to/from a network/address other than the router’s egress interface’s IP address/network.

**df-bit:** This option sets the “Don’t Fragment” bit in the IP header to indicate that routers should not fragment this packet. If it is larger than the MTU of the outbound interface, the router should drop the packet and send an ICMP Fragmentation needed and DF bit set message back to the source. This option can be very useful when you are troubleshooting MTU-related problems. By setting the df-bit option and combining it with the size option, you can force routers along the path to drop the packets if they would have to fragment them. By varying the size and looking at which point the packets start being dropped, you can determine the MTU.

```
RO1# ping 10.1.221.1 size 1476 df-bit
Type escape sequence to abort.
Sending 5, 1476-byte ICMP Echos to 10.1.221.1, timeout is 2 seconds:
Packet sent with the DF bit set
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 184/189/193 ms
RO1# ping 10.1.221.1 size 1477 df-bit
Type escape sequence to abort.
Sending 5, 1477-byte ICMP Echos to 10.1.221.1, timeout is 2 seconds:
Packet sent with the DF bit set
M.M.M
Success rate is 0 percent (0/5)
```

shows successful ping results when packet size of 1476 bytes is used; however, ping packets with a size of 1477 bytes are not successful. The **M** in the output of the ping command signifies that an ICMP Fragmentation needed and DF bit set message was received. From this, you can conclude that somewhere along the path to the destination there must be a host that has an MTU of 1476 bytes. A possible explanation for this could be usage of a generic routing encapsulation (GRE) tunnel, which typically has an MTU of 1476 bytes (1500 bytes default MTU minus 24 bytes for the GRE and IP headers).

Sweep range of sizes

```
RO1# ping
Protocol [ip]:
Target IP address: 10.1.221.1
Repeat count [5]: 1
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface:
Type of service [0]:
Set DF bit in IP header? [no]: yes
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Time stamp, Verbose[none]:
Sweep range of sizes [n]: y
Sweep min size [36]: 1400
Sweep max size [18024]: 1500
Sweep interval [1]:
Type escape sequence to abort.
Sending 101, [1400..1500]-byte ICMP Echos to 10.1.221.1, timeout is 2 seconds:
Packet sent with the DF bit set
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!M.M.M.M.M.M.M.M.M.M.M.
Success rate is 76 percent (77/101), round-trip min/avg/max = 176/184/193 ms
```

This option allows you to send a series of packets that increase in size and can be useful to determine the MTU along a path. When you want to determine the MTU of a particular path, a lot of times you do not really have a good initial guess, and it might take you many tries to find the exact MTU. In this example, the router is instructed to send packets starting at a size of 1400 bytes, sending a single packet per size and increasing the size one byte at a time until a size of 1500 bytes is reached. Again, the DF bit is set on the packets. The result is that the router sent out 101 consecutive packets, the first one was 1400 bytes, the last one was 1500 bytes, 77 of the pings

were successful, and 24 failed. Again, this means that there must be a link along the path that has an MTU of 1476 bytes.

**Note**

Because some applications cannot reassemble fragmented packets, if the network fragments that application's packet, the application will fail. Sometimes by discovering the MTU of a path, the application can be configured to not send packets larger than the MTU so that fragmentation does not happen. That is why it is sometimes necessary to find out the MTU of a path.

The various symbols generated in ping output are described as follows:

**! :** Each exclamation point indicates receipt of a reply.
**. :** Each period indicates the network server timed out while waiting for a reply.
**U :** A destination unreachable error PDU was received.
**Q :** Source quench (destination too busy).
**M:** Could not fragment.
**? :** Unknown packet type.
**& :** Packet lifetime exceeded

Telnet utility:

Telnet is an excellent companion to ping for testing transport layer connections from the command line. If you want to determine whether a particular TCP-based application is active on a server, you can attempt a Telnet connection to the TCP port of that application. For example you can use different TCP ports other than 23, for example if you need to verify the http protocol is working on the destination:

```
RO1# telnet 192.168.37.2 80
Trying 192.168.37.2, 80 ... Open
```

The Open response indicates that the port (application) you attempted is active. And the "% Connection refused by remote host" indicates that this service is not working on the destination.

You can also use the protocol name if you don't know it's TCP port number, for example:

```
RO1# telnet 192.168.37.2 smtp
```

Using Cisco IOS debug Commands

`show debugging`
Displays the state of each debugging option.

`no debug all or undebug all`
Stops all debugs.

`debug ip packet [access-list-number] detail`
this enables you to limit the scope of the **debug ip packet** command to those packets that match the access list. The **detail** option of this debug displays detailed IP packet-debugging information. This information includes the packet types and codes and source and destination port numbers.

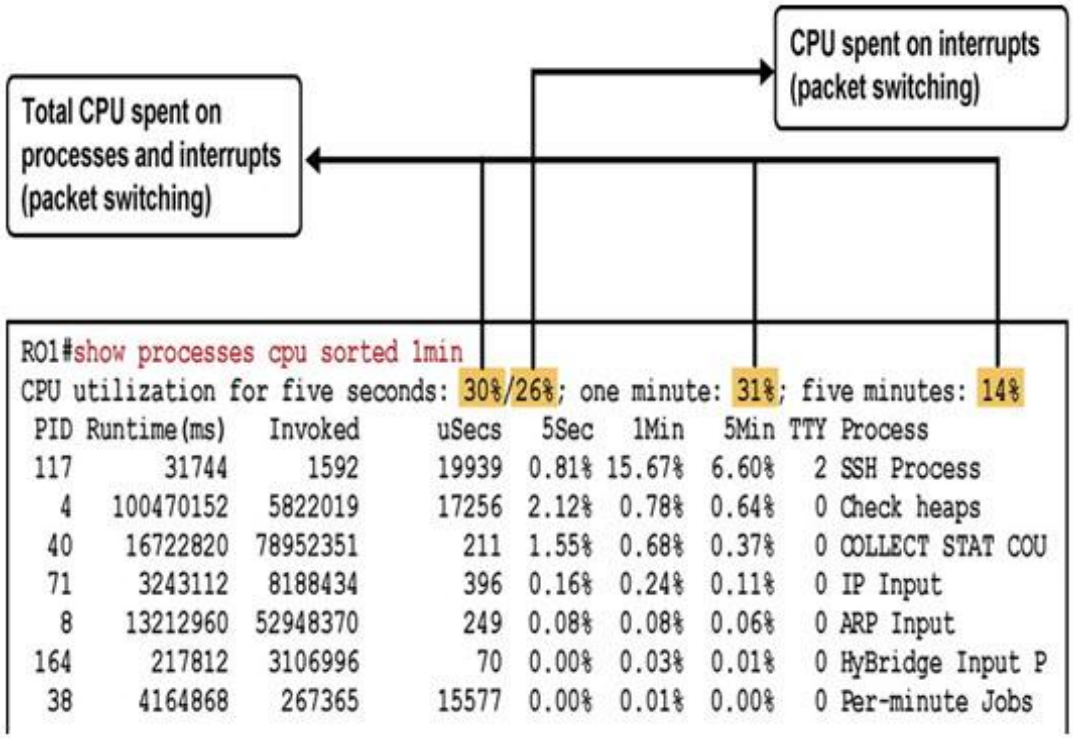
Diagnosing Hardware Issues Using Cisco IOS Commands

The three main categories of failure causes in a network are as follows: hardware failures, software failures(bugs), and configuration errors. One could argue that performance problems form a fourth category, but performance problems are symptoms rather than failure causes.

Checking CPU Utilization

Both routers and switches have a main CPU that executes the processes that constitute Cisco IOS Software. Processes are scheduled to share the available CPU cycles and take turns executing their code.

`show processes cpu`



- On the router used to generate the output depicted in, the same CPU that is used to run the operating system processes is also responsible for packet switching. The CPU is interrupted to suspend the current process that it is executing, switch one or more packets, and resume the execution of scheduled processes. The CPU time spent on interrupt-driven tasks can be calculated by adding the CPU percentages for all processes and then subtracting that total from the total CPU percentage listed at the top. For the 5- second CPU usage, this figure is actually even listed separately behind the slash. This means that in the example shown in Figure, 30 percent of the total available CPU cycles over the past 5 seconds were used, out of which 26 percent were spent in interrupt mode and 4 percent for the execution of scheduled processes.

- Because of this, it is quite normal for routers to be running at high CPU loads during peaks in network traffic. In those cases, most of the CPU cycles will be consumed in interrupt mode. If particular processes consistently use large chunks of the available CPU time, however, this could be a clue that a problem exists associated with that particular process. However, to be able to draw any definitive conclusions, you need to have a baseline of the CPU usage over time. Keep in mind that the better caching mechanisms reduce the number of CPU interrupts and, consequently, the CPU utilization attributable to interrupts. For example, Cisco Express Forwarding (CEF) in distributed mode allows most packet switching to happen on the line card without causing any CPU interrupts.

- On LAN switches, the essential elements of the show processes cpu command output are the same as routers, but the interpretation of the numbers tends to be a bit different. Switches have specialized hardware that handle the switching task, so the main CPU should in general not be involved in this. When you see a high percentage of the CPU time being spent in interrupt mode, this usually indicates that the forwarded traffic is being forwarded in software instead of by the ternary content-addressable memory (TCAM). Punted traffic is the traffic that is processed and forwarded through less-efficient means for a reason, such as tunneling or encryption. Once you have determined that the CPU load is abnormally high and you decide to investigate further, you generally have to resort to platform-specific troubleshooting commands to gain more insight into what is happening.

Checking Memory Utilization

Memory is divided into different pools and used for different purposes:

- **the processor pool:** contains memory that can be used by the scheduled processes.

- **I/O pool:** is used to temporarily buffer packets during packet switching.

```
show memory
```
If a router or switch does not have

enough free memory to satisfy the request of a process, it will log a memory allocation failure, signified by a **%SYS-2-MALLOCFAIL** message. The result of this is that the process cannot get the memory that it requires, and this might result in unpredictable disruptions or failures. Apart from the processes using up the memory through normal use, there is a possibility for memory leak. Caused by a software defect, a process that does not properly release memory (making memory to "leak" away) eventually leads to memory exhaustion and memory-allocation failures. Sometimes you need to clear the memory by reloading in no peak times.

Checking Interfaces

```
Show interfaces status or Show interfaces description
```
Shows you who is connected to whom.

```
show interfaces FastEthernet 0/0
```
The output of this command lists a number of key statistics, which are briefly described as follows:

**Input queue drops:** Input queue drops (and the related ignored and throttle counters) signify that at some point more traffic was delivered to the router than it could process. This does not necessarily indicate a problem, as it could be normal during traffic peaks. However, it may indicate that the CPU cannot process packets in time. If this number is consistently high and the dropped packets are causing application failures, the reasons must be detected and resolved.

**Output queue drops:** Input packet drops indicate congestion on the interface. Seeing output drops is normal when the aggregate input traffic rate is higher than the output traffic rate on an interface. For example the router is converting between LAN speeds and wan speed which is much lower than the LAN speed. However, even if this is considered normal behavior, it leads to packet drops and queuing delays. Applications that are sensitive to delay and packet loss, such as Voice over IP, will have serious quality issues in those situations. This counter is a good indicator that you need to implement a congestion management mechanism to provide good quality of service (QoS) to your applications.

**Input errors:** This counter indicates the number of errors such as cyclic redundancy check (CRC) errors, experienced during reception of frames. High numbers of CRC errors could indicate cabling problems, interface hardware problems, or in an Ethernet-based network, duplex mismatches. Output errors: This counter indicates the number of errors, such as collisions, during the transmission of frames. In most Ethernet-based networks today, full-duplex transmission is the norm, and half-duplex is the exception. In full-duplex operation, collisions cannot occur, and therefore collisions, and especially late collisions, often indicate duplex mismatches.

- The error counters should be evaluated against the total number of input and output packets. For example, a total of 25 CRC errors in relation to 123 input packets is reason for concern, whereas 25 CRC errors for 1,458,349 packets is not a problem at all. Furthermore, note that these counters

accumulate from the time the router boots, so the numbers displayed on the output might be accumulated over months.

- Therefore, it is difficult to diagnose a problem that has been happening over 2 days based on these statistics. After you have decided that you need to investigate the interface counters in more detail, it is good practice to reset the interface counters by using the clear counters command, let it accumulate statistics for a specific period, and then reevaluate the outcome.

- you can also Filter the Output of the show interfaces Command:

```
show interfaces FastEthernet 0/0 | include ^Fast|errors|packets
```

`show controllers`
The output of this command varies based on interface hardware type. In general, this command provides more detailed packet and error statistics for each type of hardware and interface.

`show platform`
On many of Cisco LAN switches, this command can be used to examine the TCAM and other specialized switch hardware components.

`Show hardware`
This command is equal to `show ver` command

`show inventory / raw`
This command lists the hardware components of a router or switch. The output includes the product code and serial number for each component. This is very useful for documenting your device and for ordering replacement or spare parts.

`show diag`
On routers, you can use this command to gather even more detailed information about the hardware than the output provided by the `show inventory` command. For example, the output of this command includes the hardware revision of the individual components. In case of known hardware issues, this command can be used to determine whether the component is susceptible to a particular hardware fault.

- **Generic Online Diagnostics (GOLD):** GOLD is a platform-independent framework for runtime diagnostics. It includes command-line interface (CLI)-based access to boot and health monitoring, plus on-demand and scheduled diagnostics. GOLD is available on many of the mid-range and highend Catalyst LAN switches and high-end routers such as the 7600 series and CRS-1 routers.

- **Time Domain Reflectometer.** Some of the Catalyst LAN switches support the TDR feature. This feature enables you to detect cabling problems such as open or shorted UTP wire pairs.

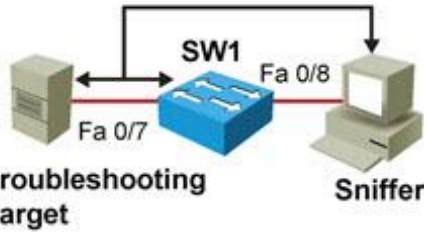
SPAN and RSPAN

**SPAN:**

```
monitor session 1 source interface Fa0/7
monitor session 1 destination interface Fa0/8
```

Sources and destinations that form a single SPAN session are identified by a session number

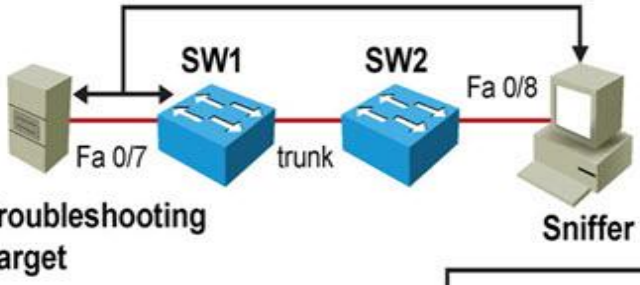
```
SW1#show monitor
Session 1
---------
Type                    : Local Session
Source Ports            :
    Both                : Fa0/7
Destination Ports       : Fa0/8
    Encapsulation       : Native
          Ingress       : Disabled
```




**RSPAN:**

```
vlan 100                                                    SW1
 name SPAN-VLAN
 remote-span
monitor session 2 source interface Fa0/7
monitor session 2 destination remote vlan 100
```

RSPAN VLAN needs to match on both switches. Session numbers do not.

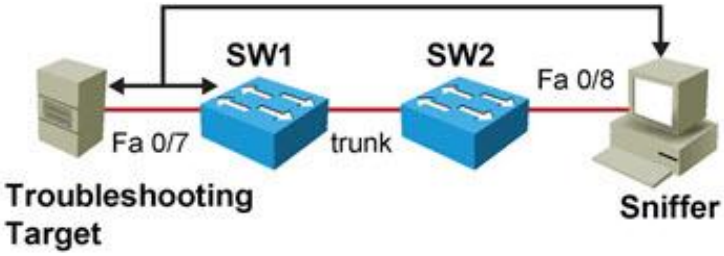



```
vlan 100                                                    SW2
 name SPAN-VLAN
 remote-span
monitor session 3 destination interface Fa0/8
monitor session 3 source remote vlan 100
```

**Verify:**

```
SW1#show monitor
Session 2
---------
Type                   : Remote Source Session
Source Ports           :
     Both              : Fa0/7
Dest RSPAN VLAN        : 100
```

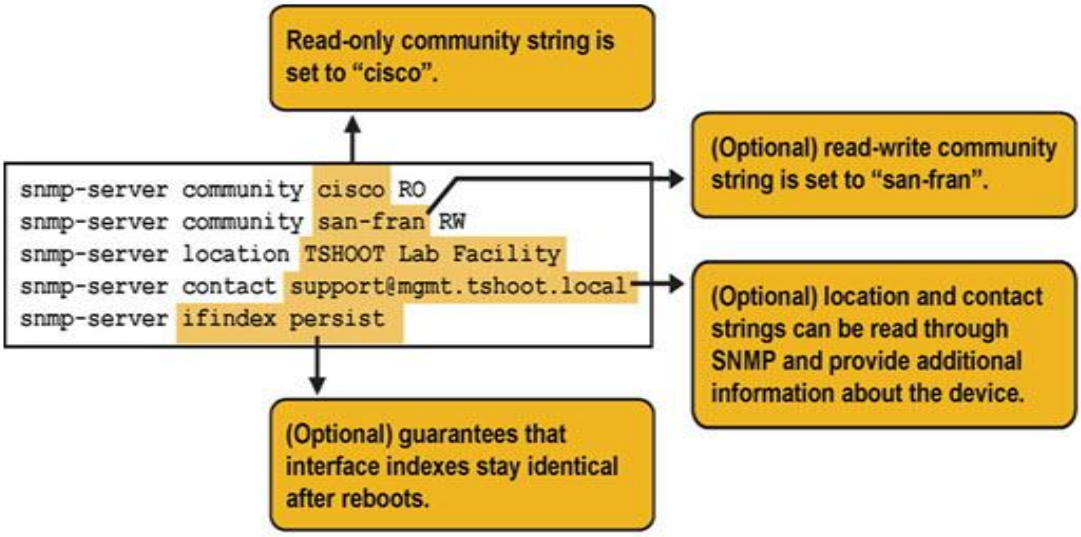
```
SW1#show vlan remote-span

Remote SPAN VLANs
------------------------
100
```




```
SW2#show vlan remote-span

Remote SPAN VLANs
------------------------
100
```

```
SW2#show monitor
Session 3
---------
Type                   : Remote Destination Session
Source RSPAN VLAN      : 100
Destination Ports      : Fa0/8
    Encapsulation      : Native
          Ingress      : Disabled
```

Gathering Information with SNMP



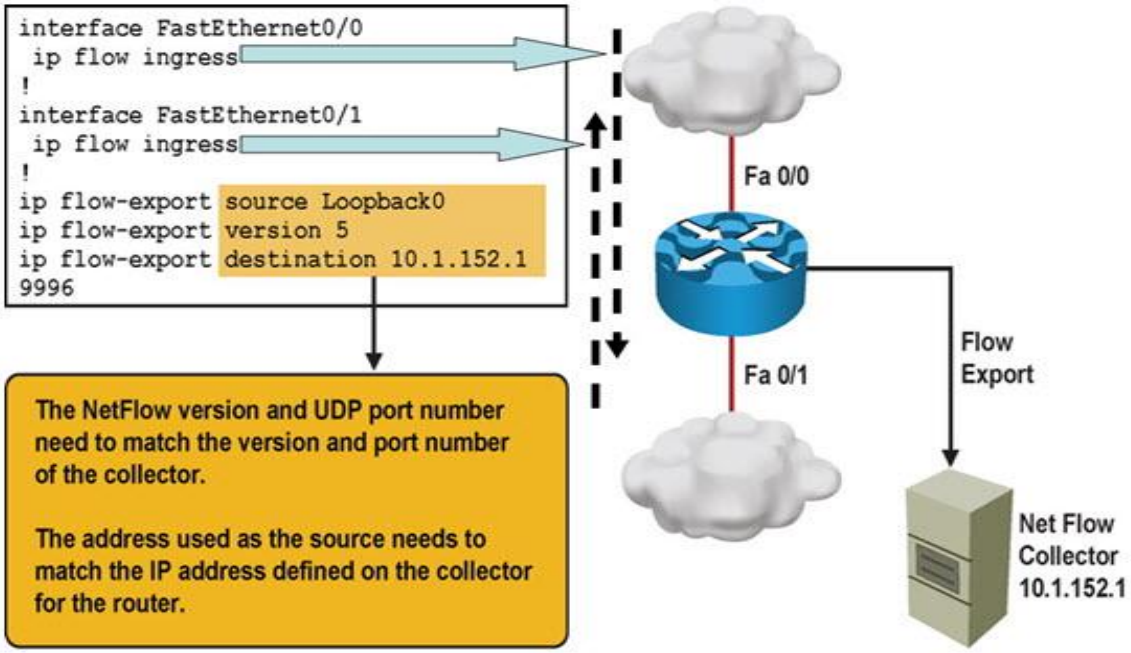

`snmp-server ifindex persist` command guarantees that the SNMP interface index for each interface will stay the same, even if the device is rebooted. Without this command, you could run into a situation where the interface's ifindex changes after a reboot and counters for that interface are no longer correctly graphed.

Gathering Information with NetFlow

NetFlow has a different focus and uses different underlying mechanisms. A NetFlow-enabled device, such as a router or Layer 3 switch, will collect information about the IP traffic that is flowing through (transit through) the device. The NetFlow feature classifies traffic by flow. A flow is identified as a collection of packets that have the same essential header fields, such as source IP address, destination IP address, protocol number, Type of Service (TOS) field, port number (if applicable), plus the ingress interface. For each individual flow, the number of packets and bytes is tracked and accounted. This information is kept in a flow cache. Flows are expired from the cache when the flows are terminated or time out.

**NOTE:**

**Netflow is now version 9.**



```
R1# show ip cache flow
<output omitted>
SrcIf         SrcIPaddress    DstIF    DstIPaddress     Pr  SrcP  DstP  Pkts
Se0/0/0.121   10.1.194.10     Null     224.0.0.10       58  0000  0000  27
Se0/0/0.121   10.1.194.14     Null     224.0.0.10       58  0000  0000  28
Fa0/0         10.1.192.5      Null     224.0.0.10       58  0000  0000  28
Fa0/1         10.1.192.13     Null     224.0.0.10       58  0000  0000  27
Fa0/1         10.1.152.1      Local    10.1.220.2       01  0000  0303   1
Se0/0/1       10.1.193.6      Null     224.0.0.10       58  0000  0000   28
Fa0/1         10.1.152.1      Se0/0/1  10.1.163.193     11  0666  E75E   1906
Se0/0/1       10.1.163.193    Fa0/0    10.1.152.1       11  E75E  0666   1905
```

```
show ip cache flow | include 10.1.163.193
```

could have been used to limit the output to only those flows that have 10.1.163.193 as the source or destination IP address.

In contrast to SNMP, NetFlow uses a "push"-based model. The collector will simply be listening to NetFlow traffic, and the routers will be in charge of sending NetFlow data to the collector, based on changes in their flow cache. Another difference between NetFlow and SNMP is that NetFlow only gathers traffic statistics, whereas SNMP can also collect many other performance indicators, such as interface errors, CPU usage, and memory usage. On the other hand, the traffic statistics collected using NetFlow have a lot more  than the traffic statistics that can be collected using SNMP.

Embedded Event Manager (EEM):

The EEM feature in Cisco IOS provides an advanced method to create custom events and define actions to be taken in response to those events. Embedded Event Manager (EEM), which enables you to define custom events and corresponding actions.

The EEM framework enables the creation of custom policies that trigger actions based on events. Events can be triggered based on various Cisco IOS subsystems such as syslog messages, Cisco IOS counter changes, SNMP MIB object changes, SNMP traps, CLI command execution, timers, and many other options. Actions can consist of sending SNMP traps or syslog messages, executing CLI commands, sending e-mail, or even running tool command language (TCL) scripts. Thus, EEM allows you to create very powerful and complex policies.

**Configuration Example:**

Assume that all network engineers within an organization are granted privileged access to the routers and switches and can make changes if necessary. The rule is that only Level 3 support engineers are allowed to make emergency changes if required, but Level 1 and 2 engineers always need to obtain authorization before making any change to the system. Whenever an engineer configures a router or switch, a %SYS-5-CONFIG_I message is logged to the syslog server. However, this message is logged as a syslog level five "notification" message and does not show up in the logs as a high-priority item. There is a requirement that a message must be logged as soon as anybody enters configuration mode; that is in addition to the %SYS-5-CONFIG_I message that is logged after configuration mode is exited. This message should be logged as a critical message and an informational message should be logged reminding the engineer of the existing change-control policies.

```
Router(config)#event manager applet CONFIG-STARTED

Router(config-applet)#event cli pattern "configure terminal" sync no skip no
occurs 1

Router(config-applet)#action 1.0 syslog priority critical msg "Configuration
mode was entered"

Router(config-applet)#action 2.0 syslog priority informational msg "Change
control policies apply. Authorized access only."
```

• The event manager applet CONFIG-STARTED command creates an applet called CONFIGSTARTED.

• The event that should trigger this applet is defined on the second line using the command event cli pattern "configure terminal" sync no skip no occurs 1. This line effectively says that the policy should be triggered if a command that includes "configure terminal" is entered. The occurs 1 option forces the event to be triggered on a single occurrence of the CLI pattern.

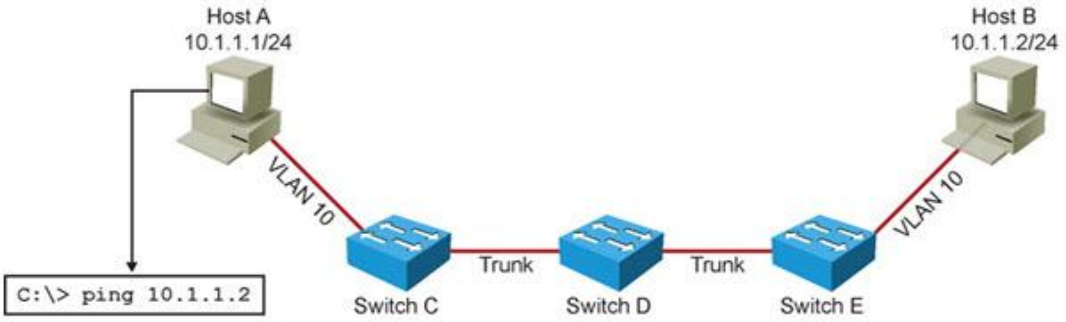
• The action 1.0 syslog priority critical msg "Configuration mode was entered" command defines an action named 1.0 (actions are sorted in alphabetic order) and instructs the router to log a critical message containing the text "Configuration mode was entered."

• The action 2.0 syslog priority informational msg "Change control policies apply. Authorized access only." command defines an action named 2.0 and instructs the router to log an informational message containing the text "Change control policies apply. Authorized access only."

```
RO1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
RO1(config)#
Mar 13 03:24:41.473 PDT: %HA_EM-2-LOG: CONFIG-STARTED: Configuration mode was
entered
Mar 13 03:24:41.473 PDT: %HA_EM-6-LOG: CONFIG-STARTED: Change control policies
apply. Authorized access only.
```

# Chapter 4. Maintaining and Troubleshooting Campus Switched Solutions

## Troubleshooting VLANs




Possible issues that could cause the communication to fail include the following:

• Physical problems

• Bad, missing, or miswired cables

• Bad ports

• Power failure

• Device problems

• Software bugs

• Performance problems

• Misconfiguration

• Missing or wrong VLANs

• Misconfigured VLAN Trunking Protocol (VTP) settings

• Wrong VLAN setting on access ports

• Missing or misconfigured trunks

• Native VLAN mismatch

• VLANs not allowed on trunk

**Note**

Note that this list is not complete, and is focused on Layer 1 and Layer 2 issues. For example, a firewall may stop the Internet Control Message Protocol (ICMP) packets. Sometimes the very first ICMP echo request times out because of the requirement for an ARP request, which is not necessary on the following ICMP echo requests.

Verifying Layer 2 Forwarding

it might be a good practice to clear the MAC entry from the table by using the:

`clear mac-address-table dynamic` command and verify that the MAC address is learned again when you reinitiate the connection.

`show mac-address-table`

This is the main command to verify Layer 2 forwarding. It shows you the MAC addresses learned by the switch and their corresponding port and VLAN associations. This command gives you an indication if frames sourced by a particular host have succeeded in reaching this switch. Note that if the MAC address table becomes full, no more learning can happen. During troubleshooting, always check to see whether the table is full.

`show vlan`

This command enables you to verify VLAN existence and port-to-VLAN associations. This command lists all VLANS that were created on the switch (either manually or through VTP). It will also list the ports that are associated to each of the VLANs. Note that trunks are not listed because they do not belong to any particular VLAN.
Also note that the vlan interface will be down down if the VLAN is not created on the switch.

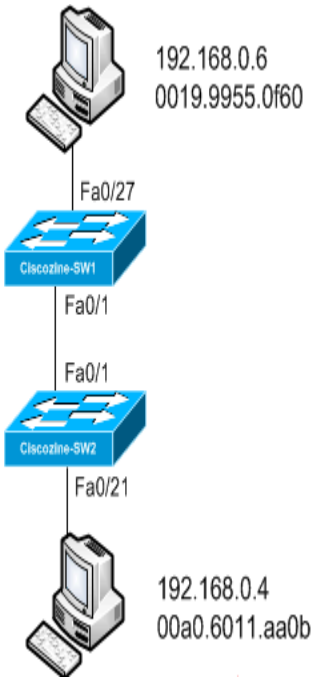
`show interfaces trunk`

`show interfaces switchport:`
shows the interface DTP related information

`show platform forward` *`interface`*
You can use many parameters with this command and find out how the hardware would forward a frame that matches the specified parameters, on the specified interface

`traceroute mac`
You specify a source and destination MAC address with this command to see the list of switch hops that a frame from that source MAC address to that destination MAC address passes through. This command requires that Cisco Discovery Protocol (CDP) be enabled on all the switches in the network (or at least within the path).

```
Ciscozine-SW1#traceroute mac 0019.9955.0f60 00a0.6011.aa0b

Source 0019.9955.0f60 found on Ciscozine-SW1
1 Ciscozine-SW1 (192.168.0.253) : Fa0/21 => Fa0/1
2 Ciscozine-SW2 (192.168.0.254) : Fa0/1 => Fa0/27
Destination 00a0.6011.aa0b found on Ciscozine-SW2
Layer 2 trace completed
Ciscozine-SW1#
```




If you do not know the mac address use the following command:

```
Ciscozine-SW1#traceroute mac ip 192.168.0.4 192.168.0.6
Translating IP to mac .....
192.168.0.4 => 00a0.6011.aa0b
192.168.0.6 => 0019.9955.0f60

Source 00a0.6011.aa0b found on Ciscozine-SW2
1 Ciscozine-SW2 (192.168.0.254) : Fa0/27 => Fa0/1
2 Ciscozine-SW1 (192.168.0.253) : Fa0/1 => Fa0/21
Destination 0019.9955.0f60 found on Ciscozine-SW1
Layer 2 trace completed
```

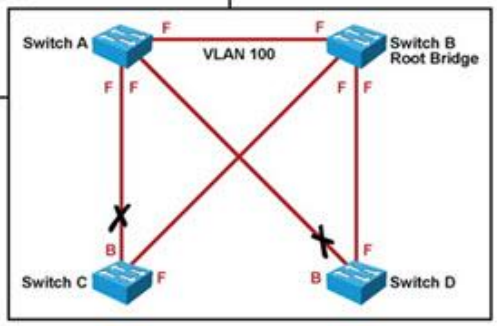
## Troubleshooting Spanning Tree

Analyzing the Spanning-Tree Topology:

- The original spanning-tree timers are based on the assumption that the network diameter is up to seven bridges/switches long, "connected as daisy chain".

- show spanning-tree [vlan *vlan-id*]

```
SwitchA#show spanning-tree interface Ten 9/1 detail
 Port 88 (TenGigabitEthernet9/1) of VLAN0100 is root forwarding
   Port path cost 2, Port priority 128, Port Identifier 128.88.
   Designated root has priority 28772, address 0000.0c9f.3127
   Designated bridge has priority 28772, address 0000.0c9f.3127
   Designated port id is 128.88, designated path cost 0
   Timers: message age 15, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default
   BPDU: sent 10, received 670
```




- show spanning-tree interface interface-id detail

In the example shown in Figure, you can see that port 88 (TenGigabitEthernet9/1) is a root port and the upstream switch's port is the designated port. This is also reflected by the fact that this switch is receiving BPDUs (it received 670 BPDUs), but not transmitting them (it sent 10 BPDUs during initial spanning-tree convergence and stopped after that). You can also see that the upstream switch is the root bridge. This can be concluded from the fact that the designated bridge ID and the root bridge ID are the same. This is further confirmed by the fact that the designated path cost is reported as a cost of 0.

**Note:**

This statement "**Designated bridge has priority**" means that the upstream switch is the designated switch.

Spanning-Tree Failures:

An Ethernet frame header does not include a Time To Live (TTL) field; therefore, any frame that enters a bridging loop will continue to be forwarded by the switches indefinitely. The only exceptions are the frames that have their destination address recorded in the MAC address table of the switches. These frames will simply be forwarded to the port that the MAC address is associated with and will not go into an endless loop. However, any frame that is flooded by a switch, such as broadcasts, multicasts, and unicasts with an unknown destination MAC address, will go into an

endless loop and start circling. The consequences and corresponding symptoms of this behavior are as follows:

• The load on all links in the switched LAN will quickly start increasing as more and more frames enter the loop. Note that this is not limited to just the links that form the loop, but also any other links in the switched domain, because some frames are flooded on all links. Naturally, when the spanning tree failure is limited to a single VLAN, then only links in that VLAN will be affected and switches and trunks that do not carry that VLAN will operate normally.

• If the spanning-tree failure has caused more than one bridging loop to form, traffic will increase exponentially (because frames will not only start circling, but will also start getting duplicated). This happens because, in the case of multiple loops, there will be switches that receive a frame on a port and then flood it out on multiple ports, essentially creating a copy of the frame every time they forward it.

• When control plane traffic, such as Hot Standby Router Protocol (HSRP), OSPF, or Enhanced Interior Gateway Routing Protocol (EIGRP) hellos, starts entering the loop, the devices that are running these protocols will quickly start getting overloaded. Their CPU will approach 100 percent utilization while they are trying to process an ever-increasing load of control plane traffic. In many cases, the earliest indication of a broadcast storm in progress is that routers or Layer 3 switches are reporting control plane failures, such as continual HSRP state changes, or that they are running at a very high CPU utilization load.

• Switches will experience very frequent MAC address table changes. This happens because frames might loop in both directions, causing a switch to see a frame with a certain source MAC address enter through one port and then see a frame with the same source MAC address enter through a different port shortly later.

• Because of the combination of very high load on all links and the CPU running at maximum load on Layer 3 switches or routers, these devices typically become unreachable, making it nearly impossible to diagnose the problem while it is in progress.

One intrusive but effective way to start tackling severe spanning-tree problems is eliminating topological loops by either physically disconnecting links or by shutting down interfaces if that is still possible. Once the loops are broken, the traffic and CPU loads should quickly drop to normal levels, and you should regain connectivity to your devices. Although this restores connectivity to the network, you cannot consider this the end of your troubleshooting process. You have removed all redundancy from your switched network, and you need to restore the redundant links.

**EtherChannel Operation:**

There are three common EtherChannel problems:

**Inconsistencies between the physical ports that are members of the channel:**

The physical links in an EtherChannel must have the same operational characteristics. For example, they must have the same speed, duplex, trunk, or access port status, native VLAN when trunking, and same access VLAN when they are access ports. the switch suspends a physical link in the channel because of incompatibilities, it generates a

`%EC-5-CANNOT_BUNDLE2` log message.

**Inconsistencies between the ports on the opposite sides of the EtherChannel link:**

If the switch on one side of a few links is configured to bundle these links into an EtherChannel and the switch on the other side is not, the switch that is configured for EtherChannel will detect this (by detecting inconsistencies in the spanning-tree behavior) and move the port to an error-disabled state. The switch will generate a %SPANTREE-2-CHNL_MISCFG message when it "error disables" the port. The use of an EtherChannel negotiation protocol like the 802.3ad Link Aggregation Control Protocol (LACP) or the Port Aggregation Protocol (PAgP) prevents this situation from happening because both sides must first agree to form the channel.

**NOTE:**

**Uneven distribution of traffic between EtherChannel bundle members:**

Some people expect that when EtherChannel is used, the traffic is equally balanced across all physical links in the bundle. You must realize, however, that the method used to distribute traffic over the physical links is to calculate a hash of a combination of fields in the Ethernet and IP headers of a frame and then send the frame to a physical interface based on the hash result. Therefore, the distribution of traffic depends on two things: The distribution of hash values over the physical links, and the header fields that are used as a key into the hash calculation. The Cisco EtherChannel hash algorithm results in a value between 0 and 7. This means that in case of an eight-port EtherChannel, one hash value is assigned to each of the links, and (assuming a random traffic mix) traffic is equally balanced across all eight links. However, if the channel consists of six links, the distribution will be 2:2:1:1:1:1 instead, meaning that the first two links in the channel will each handle twice as much traffic as the other links. The second factor in EtherChannel load balancing is which header fields are used as the base of the hash value. If you could assume those fields in the traffic to be entirely random, it would not matter what hashing mechanism were used; however, because header fields are typically not random, the choice of header fields to be hashed does affect the distribution. For example, when only the destination MAC address is used as the input for the hash calculation, if 90 percent of all frames are destined for a single MAC address

(for instance, the MAC address of the default gateway), all of that traffic would end up on the same physical link. Therefore, if you see an uneven distribution of traffic over the links in the channel, you should examine the hashing method and the traffic mix to determine the cause.

**Troubleshooting Example:**

A broken access switch has been replaced by a new access switch ASW1. The junior support staff has configured ASW1 to the best of his knowledge and using the documentation that exists. After the switch booted and its physical connections to the two other switches (CSW1 and CSW2) were restored, the junior support staff reported three problems that he could not solve:

• On CSW2, port channel 1, which connects to ASW1, is down.

• On ASW1, the following log message on the console indicates a spanning-tree problem on Po2, which connects to CSW1: %SPANTREE-2-PVSTSIM_FAIL: Blocking designated port Po2: Inconsistent superior PVST BPDU received on VLAN 17, claiming root 24593:001f.2721.8400

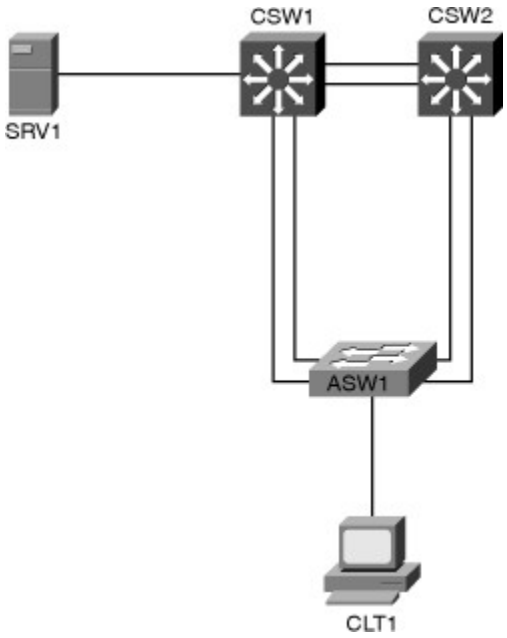
• On ASW1, interface VLAN 128 is down.

```
CSW2# show etherchannel summary
```

```
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SD)           -        Fa0/5(s)         Fa0/6(s)
2      Po2(SU)           -        Fa0/3(P)         Fa0/4(P)
```

S: means L2 Etherchannel.
s: means suspended.

When you see physical interfaces in an EtherChannel that are marked as suspended, this usually indicates that a configuration mismatch, either between interfaces in the channel itself or between the configuration on this end of the EtherChannel and the configuration at the other end.




```
ASW1# show spanning-tree blockedports
```

```
ASW1# show spanning-tree root
```

To find more detail about what exactly caused the problem, you can use the following command:

```
CSW2# show etherchannel 1 detail
Group state = L2
Ports: 2 Maxports = 8
Port-channels: 1 Max Port-channels = 1
Protocol: -
Minimum Links: 0
Ports in the group:
----------
Port: Fa0/5
------
Port state = Up Cnt-bndl Suspend Not-in-Bndl
Channel group = 1 Mode = On Gcchange = -
Port-channel = null GC = - Pseudo port-channel = Po1
Port index = 0 Load = 0x00 Protocol = -
Age of the port in the current state: 0d:00h:25m:13s
Probable reason: vlan mask is different
Port: Fa0/6
------
Port state = Up Cnt-bndl Suspend Not-in-Bndl
Channel group = 1 Mode = On Gcchange = -
Port-channel = null GC = - Pseudo port-channel = Po1
Port index = 0 Load = 0x00 Protocol = -
Age of the port in the current state: 0d:00h:25m:14s
Probable reason: vlan mask is different
```

The output shown in indicates that the cause of the problem is the VLAN mask, which means that there must be a mismatch between the VLANs allowed on the port channel versus the VLANs allowed on the physical interfaces.

You could also find the problem indication from the log, which contains the messages

```
Mar 20 08:12:39 PDT: %EC-5-CANNOT_BUNDLE2: Fa0/5 is not compatible with Po1 and
will be suspended (vlan mask is different)
Mar 20 08:12:39 PDT: %EC-5-CANNOT_BUNDLE2: Fa0/6 is not compatible with Po1 and
will be suspended (vlan mask is different)
```

**Solution:**

These findings lead you to compare the port-channel interface to the physical interfaces to find out that the VLAN allowed list is missing on physical interfaces Fa0/5 and Fa0/6 of ASW1.

Investigate the second problem; use the `show spanning-tree` command on VLAN 17:

```
ASW1# show spanning-tree vlan 17

MST0
  Spanning tree enabled protocol mstp
  Root ID     Priority    32768
              Address     001e.79a9.b580
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32768  (priority 32768 sys-id-ext 0)
              Address     001e.79a9.b580
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/7               Desg FWD 200000    128.9    P2p Edge
Po1                 Desg BLK 100000    128.56   P2p
Po2                 Desg BKN*100000    128.64   P2p Bound(PVST) *PVST_Inc
```

The output of this command has two elements that clearly point to a spanning-tree configuration issue. The BKN* and *PVST_Inc elements in the output point toward a spanning-tree inconsistency, while the Bound (PVST) element points toward a boundary between two different spanning-tree varieties. Because all other switches run Rapid Per-VLAN Spanning Tree (R-PVST+), it is reasonably safe to assume that switch ASW1 should not be running Multiple Spanning Tree (MST), but should be running R-PVST+. Note that the third line clearly indicates that ASW1 is running MST.

Investigating the third problem; vlan 128 is down:

```
ASW1# show ip interfaces brief | exclude unassigned
Interface              IP-Address      OK? Method Status                Protocol
Vlan128                10.1.156.1      YES NVRAM  up                    down
```

A VLAN interface is up as long as the VLAN exists and there is an active port in that VLAN that is in spanning tree forwarding state. Therefore, when you discover that a VLAN interface is down, it is a good idea to first check the spanning-tree status for that VLAN.

```
ASW1# show spanning-tree vlan 128
Spanning tree instance(s) for vlan 128 does not exist.

ASW1# show vlan id 128
VLAN id 128 not found in current VLAN database
```

## Troubleshooting Switched Virtual Interfaces and Inter-VLAN Routing

Differences Between Multilayer Switches and Routers:

1- Routers connect heterogeneous networks and support a wide variety of media and interfaces. Multilayer switches typically connect homogenous networks. Nowadays LAN switches are mostly Ethernet only.

2- Multilayer switches use specialized hardware to achieve wire-speed Ethernet-to-Ethernet packet switching. Low- to mid-range routers use multipurpose hardware to perform the packet-switching process. On average, the packet-switching throughput of routers is lower than the packet-switching throughput of multilayer switches.

3- Routers usually support a wider range of features, mainly because switches need specialized hardware to be able to support certain data plane features or protocols. On routers, you can often add features through a software update.

**Note:**

- Remember that when using SVIs on Multi-layer switches, you need to enable ip routing using the following command: Sw(config)#ip routing

- Traditionally on Layer 2 switches, the term **port** has been used, and on routers the term **interface** has been used.

**- An SVI will be up up according to these conditions:**

1- At least one port associated to the corresponding VLAN; that port has to be up and in the spanning-tree forwarding state.

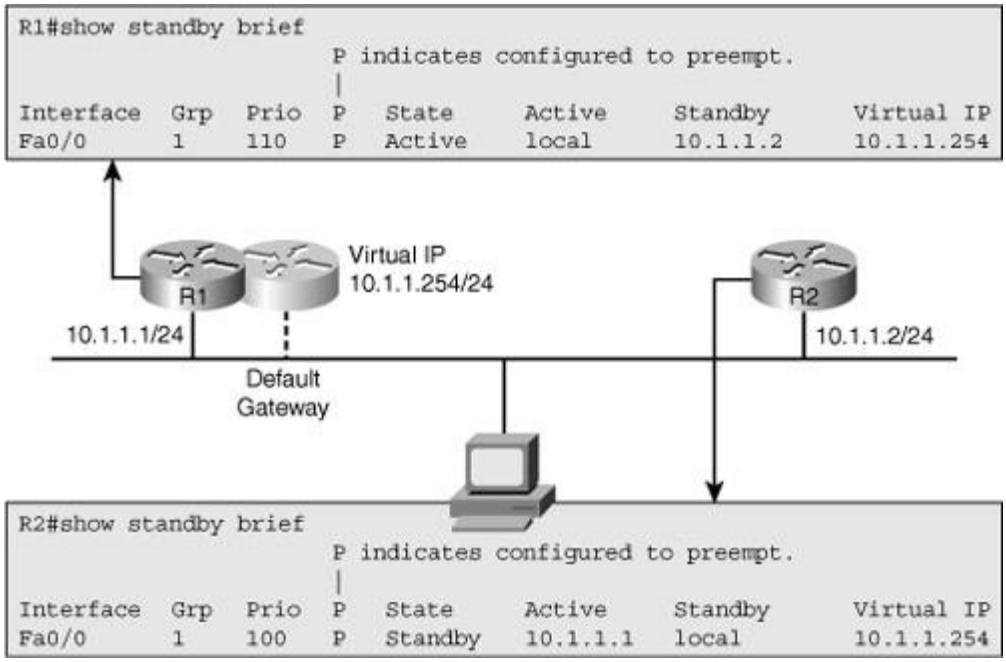
2- Note that this rule includes both access ports and trunks that have this VLAN in their allowed VLAN list.

3- The VLAN itself should be existing on the switch.

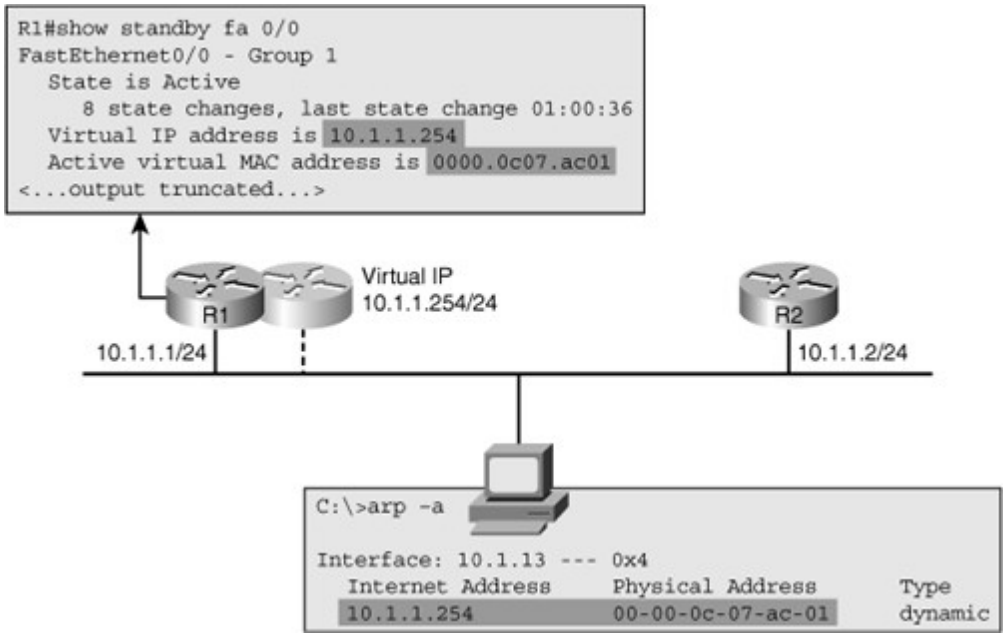
As a result, an SVI can only go down when the last active port in the VLAN goes down or loses its spanning-tree forwarding status.

## Verifying FHRP Operation:

`show standby brief`

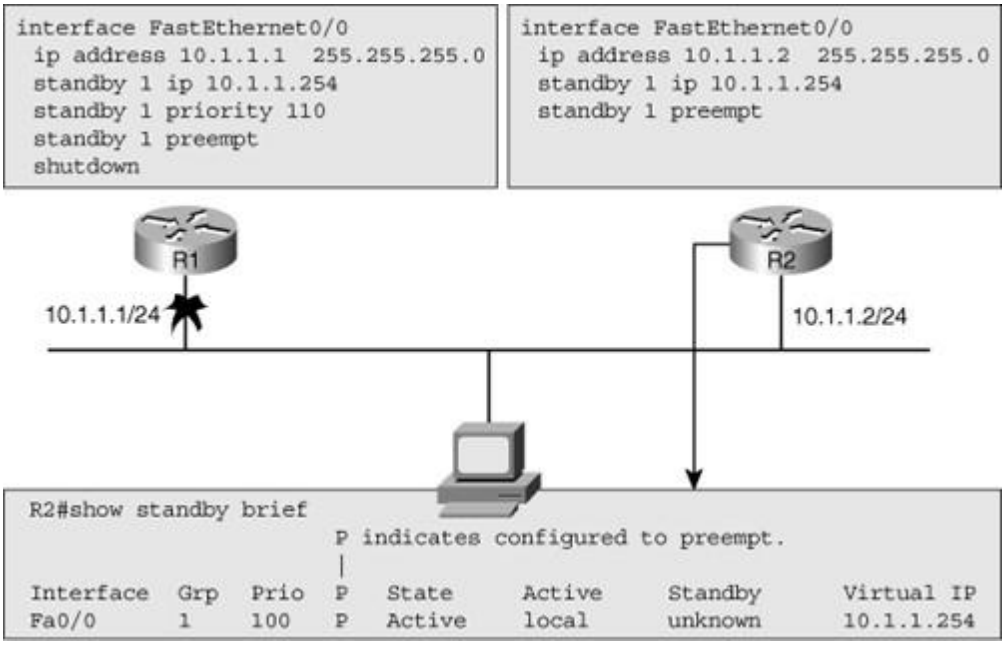



`show standby` *`interface id`*



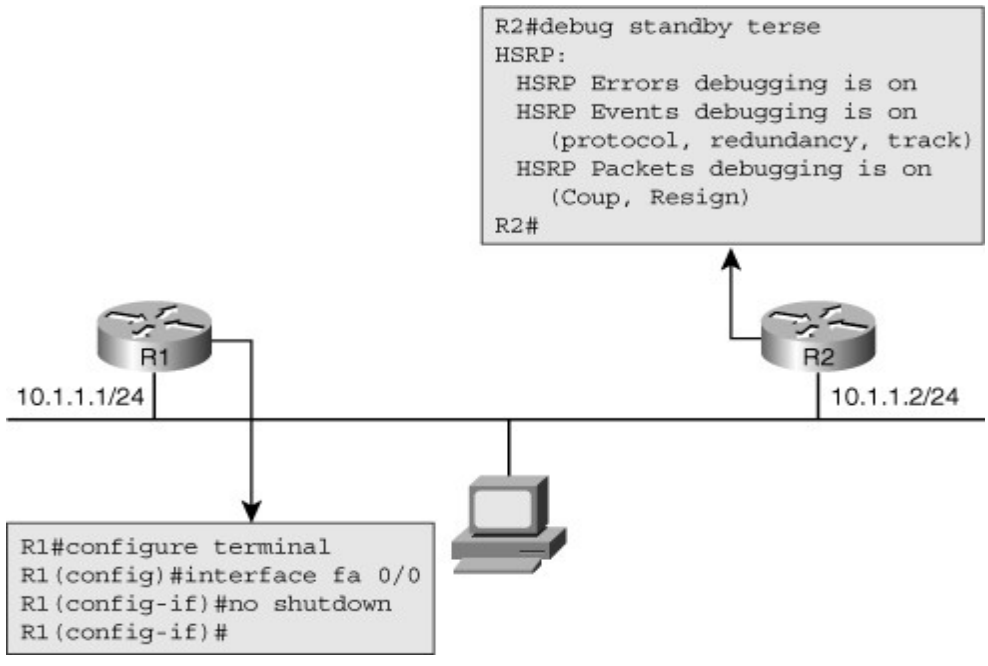
- When you are troubleshooting HSRP-related problems, it is useful to know the virtual MAC address used for the standby group because it can be used to verify the correct operation of ARP and the Layer 2 connectivity between the end host and the active HSRP router.

- In many cases, HSRP-related problems are not really, at the root, caused by HSRP itself, but by problems in the underlying switched network. For example, a typical symptom of a broadcast storm is that you start seeing frequent HSRP state changes on the Layer 3 switches that are connected to the affected VLANs.




- This issue shows that R2 does not detect R1 as a standby group member, in other words it does not connected directly to R1.

- Another issue is password or key chain misconfigurations if you use FHRP authentication.



- debug standby terse

This is a good debugging command to start with because it includes most of the relevant messages, but excludes the HSRP hellos, keeping the output of the debug limited and readable.

- When R1 comes up on the segment, because it has a higher priority than the current active router and it is configured with the preempt option, it sends out a "coup" message to take over the active role.

- If R1 is for example reloading or its f0/0 is shutdown, it will send a "resign" message telling R2 that it will leave.

**VRRP:**

- VRRP is an IETF standard (RFC 3768), which makes it suitable for multivendor environments. HSRP and GLBP are a CISCO protocols.

- It preempts by default, but HSRP and GLBP does not

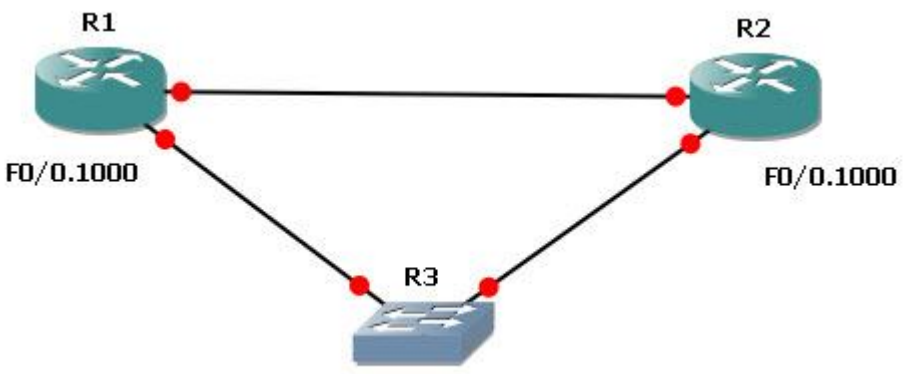
**Table 4-2. Operational Differences Between HSRP, VRRP, and GLBP**

| Feature | HSRP | VRRP | GLBP |
|---|---|---|---|
| Transparent default gateway redundancy. | Yes | Yes | Yes |
| Virtual IP address can also be a real address. | No | Yes | No |
| IETF standard. | No | Yes | No |
| Preempt is enabled by default. | No | Yes | No |
| Multiple active forwarding gateways per group. | No | No | Yes |
| Default hello timer (seconds). | 3 | 1 | 3 |

**Table 4-3. Main Troubleshooting Commands for HSRP, VRRP, and GLBP**

| HSRP | VRRP | GLBP |
|---|---|---|
| **show standby brief** | **show vrrp brief** | **show glbp brief** |
| **show standby***interface-id* | **show vrrp interface** *interface-id* | **show glbp** *interface-id* |
| **debug standby terse** | No real equivalent option exits. Multiple debug options must be used simultaneously. | **debug glbp terse** |

**NOTE:**



If you use 802.1Q on the router sub-interface for example f0/0.1000 and in the same time the native vlan is 1000, when R3 receives a packet with the tag of 1000, it will remove the tag before forwarding the packet To R2, it will assume all FHRP protocols packets as a native vlan packets, so R1 and R2 will not see each other in FHRP, so the native vlan should be something other than vlan 1000.

# Chapter 5. Maintaining and Troubleshooting Routing Solutions

**Notes:**

- when a host tries to communicate with other host, first I inspects if the other host is on the same subnet by comparing the ip address and the subnet mask. If the other host on the same subnet, the host tries to get the mac address using ARP, if on another subnet the host sends the packets to the default gateway, but first it uses ARP to get the MAC address of the default gateway.

- Router C decrements the Time To Live (TTL) field in the IP header of the packet by one. If this causes the TTL field to be set to zero, Router C will discard the packet and send an Internet Control Message Protocol (ICMP) "time exceeded" message back to the source, Host A. If the TTL of the packet is not reduced to zero the router performs a forwarding table lookup to find the longest prefix that matches the destination IP address of the packet being processed.

- Remember that no mac address are used on WAN.

**- Cisco Express Forwarding [CEF]:**

*- Cisco Express Forwarding (CEF) is an advanced layer 3 switching technology used mainly in large core networks or the Internet to enhance the overall network performance. Although CEF is a Cisco proprietary protocol other vendors of multi-layer switches or high-capacity routers offer a similar functionality where layer-3 switching or routing is done in hardware (in an ASIC) instead of by software and the (central) (CPU).

*- CEF is mainly used to increase packet switching speed by reducing the overhead and delays introduced by other routing techniques. CEF consists of two key components: **The Forwarding Information Base (FIB)** and **adjacencies**.

The **FIB** is similar to the routing table generated by multiple routing protocols, maintaining only the next-hop address for a particular IP-route.

The **adjacency table** maintains layer 2 or switching information linked to a particular FIB entry, avoiding the need for an ARP request for each table lookup. There are several types of adjacencies. Some are listed below:

**Cache adjacency**: This type of entry contains the correct outbound interface and the correct MAC address for its FIB entry. The MAC address is the IP address's MAC address if the destination's subnet is directly connected to the router, or is the MAC address of the router that the packet needs

to be sent to if the destination's subnet is not directly connected to the router currently processing the packet.

**Note:**

Serial interfaces does not have MAC address, this case describes the router's Ethernet interfaces, or mainly the switches interfaces.

**Receive adjacency**: This type of entry handles packets whose final destinations include the router itself. This includes packets whose IP addresses are assigned to the router itself, broadcast packets, and multicasts that have set up the router itself as one of the destinations.

**Null adjacency:** Handles packets destined to a NULL interface. Packets with FIB entries pointing to NULL adjacencies will normally be dropped.

**Punt adjacency:** Deals with packets that require special handling or can not be switched by CEF. Such packets are forwarded to the next switching layer (generally fast switching) where they can be forwarded correctly.

**Glean adjacency**: This adjacency is created when the router knows that either the destination IP's subnet is directly connected to the router itself and it does not know that destination device's MAC address, or the router knows the IP address of the router to forward a packet to for a destination, but it does not know that router's MAC address. Packets that trigger this entry will generate an ARP request.

**Discard adjacency**: FIB entries pointing to this type of adjacency will be discarded.

**Drop adjacency**: Packets pointing to this entry are dropped, but the prefix will be checked.

In order to take full advantage of CEF, it is recommended to use distributed CEF (dCEF), where there is a FIB table on each of the line cards. This avoids the need for querying the main processor or routing table in order to get the next-hop information. Instead, fast switching will be performed on the line card itself.

**CEF currently supports Ethernet, Frame Relay, ATM, PPP, FDDI, tunnels, and Cisco HDLC.**

Using IOS Commands to Verify Routing Functions

To determine the information that is used to forward packets, you can verify the availability of specific routing entry (prefix) in the routing table or the CEF FIB table.

```
Show ip route
Show ip route 192.168.10.1
Show ip route 192.168.10.0 255.255.255.0
Show ip route 192.168.10.0 255.255.255.0 longer-prefixes
```
This command can prove useful to diagnose problems related to route summarization.

`show ip cef` *`ip-address`*
This command is similar to the `show ip route ip-address` command, but it searches the FIB rather than the routing table. Therefore, the displayed results do not include any routing protocol-related information, but only the information necessary to forward packets. (Note that this command will display the default route if it is the best match for a particular IP address.)

`show ip cef network mask`
This is similar to the `show ip route network mask` command, but it displays information from the FIB rather than the routing table (RIB).

`show ip cef exact-route` *`source destination`*
This command displays the exact adjacency that will be used to forward a packet with source and destination IP addresses, as specified by the source and destination parameters. The main reason to use this command is in a situation when you are tracking a packet flow across the routed network but the routing table and FIB contain two or more equal routes for a particular prefix. In this case, the CEF mechanisms will balance the traffic load across the multiple adjacencies associated with that prefix. By use of this command, you can determine which of the possible adjacencies is used to forward packets for a specific source and destination IP address pair.

```
Show ip arp

Clear ip arp

Show frame-relay map
```

```
show adjacency detail
```

When CEF is used as the switching method, the information from the various Layer 2 data structures is used to construct a frame header for each adjacency that is listed in the adjacency table. You can display the full frame header that will be used to encapsulate the packet

## Troubleshooting EIGRP

EIGRP stores its operational data, configured parameters, and statistics in three main data structures:

**Interface table:** This table lists all interfaces that have been enabled for the processing of EIGRP packets, such as hellos, updates, queries, replies, and acknowledgments. Passive interfaces are not listed in this table.

**Neighbor table:** This table is used to keep track of all active EIGRP neighbors. Neighbors are added to this table based on the reception of hello packets, and they are removed when a neighbors holdtime expires or when the associated interface goes down or is removed from the interface table. This table is also used to keep track of the status of any routing information exchanges between the router and its neighbors.

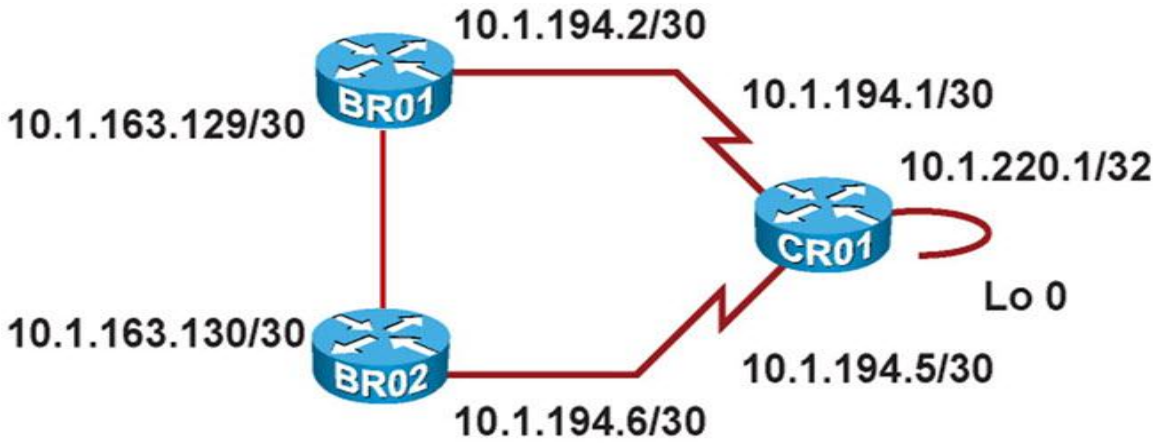
**Topology table:** This table holds all the routes that were received from neighboring routers, locally injected, or redistributed into EIGRP. For each prefix, EIGRP selects the best path from among the available possible paths stored in this table to be offered to the IP routing table. EIGRP's best path selection is based on the diffusing update algorithm (DUAL). If multiple paths have the exact same metric, all entries that share that same metric are selected for installation in the routing table. Routes with a higher metric are not selected for installation in the routing table, unless unequal-cost load balancing has been enabled.

**Routing Table:** the best Routes "lowest cost" are stored in this table.

Monitoring EIGRP

```
show ip eigrp interfaces
show ip eigrp neighbors
show ip eigrp topology
debug ip routing
debug eigrp packets
debug ip eigrp
debug ip eigrp neighbor as-number ip-address
```

**Troubleshooting Example: Routing Problem in an EIGRP Network**




When a traceroute command is executed on the BRO1 router to determine the path used to reach the loopback of router CRO1, as shown in Example, it turns out that the traffic path goes through BRO2.

```
BRO1# traceroute 10.1.220.1

Tracing the route to cro1.mgmt.tshoot.local (10.1.220.1)

1 10.1.163.130 0 msec 0 msec 0 msec
2 10.1.194.5 12 msec 12 msec *
```

The first reason could be that router BRO1 has not learned about the direct route to router CRO1. The second reason could be that the router has learned about the route but incorrectly selects the route through router BRO2 as the best route.

```
BRO1# show ip eigrp topology 10.1.220.1 255.255.255.255
IP-EIGRP (AS 1): Topology entry for 10.1.220.1/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 40642560
  Routing Descriptor Blocks:
  10.1.163.130 (FastEthernet0/1.30), from 10.1.163.130, Send flag is 0x0
      Composite metric is (40642560/40640000), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 25100 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
```

only one entry is listed in BRO1's topology table, you must now determine whether the route was not learned because a neighbor relationship with CRO1 was never established or if the relationship was established but the specific route was not exchanged.

```
BRO1# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.163.130            Fa0/1.30          12 00:09:56    4   200  0  585
```

So one reason that could explain why router CRO1 is not listed is that router BRO1 has not received any hello packets from router CRO1. Another explanation could be that the packets were received, but ignored, because router BRO1 is not processing EIGRP packets on that interface. The interface table will allow you to see which interfaces are enabled for EIGRP processing. Example shows the output of the show ip eigrp interfaces command on router BRO1.

```
BRO1# show ip eigrp interfaces
IP-EIGRP interfaces for process 1

                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Fa0/1.30           1        0/0         4       0/1           50           0
```

Two conditions need to be met for an interface to be added to the EIGRP interface table:

• The interface has to be up, and its IP address must match one of the configured network statements.

• The interface should not be configured as a passive interface.

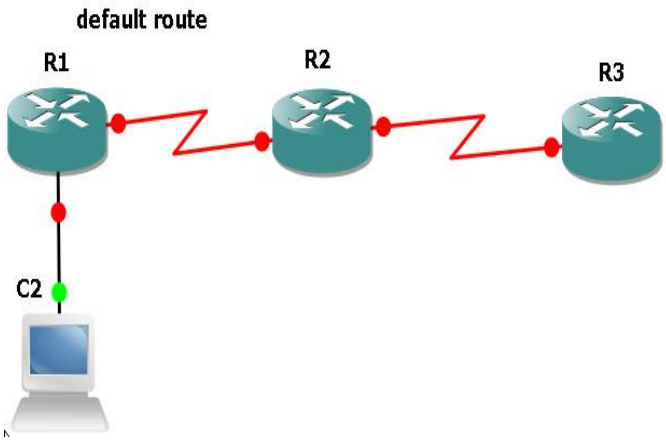
```
BRO1# show running-config | section router eigrp
router eigrp 1
network 10.1.163.129 0.0.0.0
network 10.1.194.1 0.0.0.0
no auto-summary
```

a problem exists with one of the network statements. The statement network 10.1.194.1 0.0.0.0matches IP address 10.1.194.1, which is not one of router BRO1's IP addresses, but an IP address of router CRO1. This is clearly a configuration mistake, and the network statement needs to be replaced with the statement network 10.1.194.2 0.0.0.0 or some other network statement that matches IP address 10.1.194.2, which is the IP address of router BRO1 for the WAN link.

```
BRO1# traceroute 10.1.220.1
Tracing the route to cro1.mgmt.tshoot.local (10.1.220.1)
1 10.1.194.1 16 msec 12 msec *
```

**Note:**

R1 has ad default route to R2, so when the host tries to reach R3 networks R1 will send the packet to the default "R2", if R2 does not know how to reach R3 networks it will send back a "Unreachable" message to R1, but R1 will send back a "Timed out" message to the host.



## Troubleshooting OSPF

OSPF stores its operational data, configured parameters, and statistics in four main data structures:

**- Interface table.**

**- Neighbor table.**

**- Link-state database.**

**- Routing Information Base:** After executing the SPF algorithm, the results of this calculation are stored in the RIB. This information includes the best routes to each individual prefix in the OSPF network with their associated path costs. When the information in the link-state database changes, only a partial recalculation might be necessary (depending on the nature of the change), and routes might be added to or deleted from the RIB without the need for a full SPF recalculation. From the RIB, OSPF offers its routes to the IP routing table.

## Note

Within the OSPF link-state database, the best path to each destination is determined based on the SPF (Dijkstra) algorithm. The collection of these best paths is referred to as the OSPF RIB. There is no separate physical data structure called the OSPF RIB. The best path to each destination is offered to be installed in the IP routing table. When there are alternatives, the IP process selects the path with the smallest administrative distance and installs it in the IP routing table. As of year 2001 and after the release of RFC 3222, many writings have referred to the IP routing table as the RIB (generic RIB rather than OSPF or BGP RIB). This term is easier to use than IP routing table, and it allows us to distinguish it from the FIB that CEF creates. Although the FIB is created based on RIB, it is indeed a separate data structure, and IP packet forwarding (a data plane task) in a Cisco router is performed using FIB and the FIB adjacency table.

## OSPF Information Flow Within an Area

OSPF discovers neighbors through the transmission of periodic Hello packets. Two routers will become neighbors only if the following parameters match in the Hello packets:

• **Hello and dead timers:** Two routers will only become neighbors if they use the same Hello and dead time. The default values for broadcast and point-to-point type networks are 10-second Hello and 40-second dead time. If these timers are changed on an interface of a router, the timers should be configured to match on all neighboring routers on that interface.

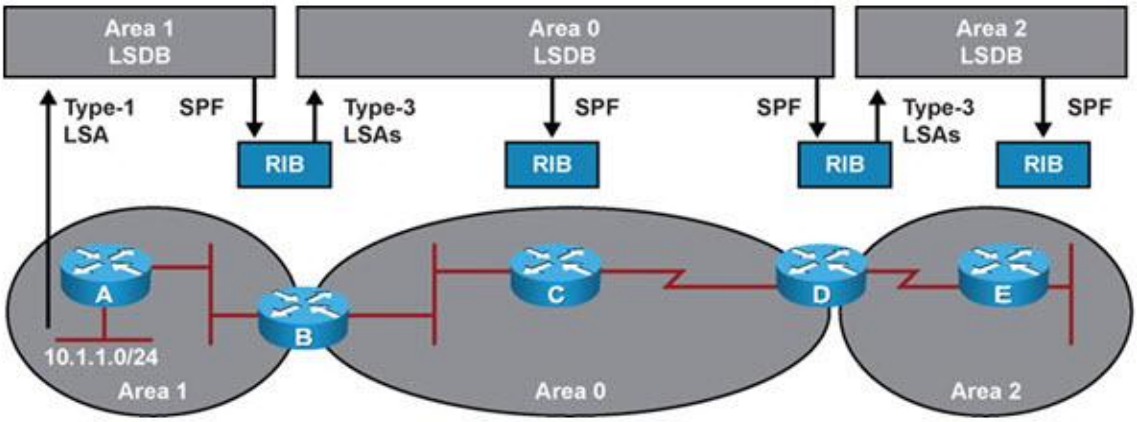
• **OSPF area number:** Two routers will become neighbors on a link only if they both consider that link to be in the same area.

• **OSPF area type:** Two routers will become neighbors only if they both consider the area to be the same type of area (normal, stub, or not-so-stubby area [NSSA]).

• **IP subnet and subnet mask:** Two routers will not become neighbors if they are not on the same subnet. The exception to this rule is on a point-to-point link, where the subnet mask is not verified.

• **Authentication type and authentication data:** Two routers will become neighbors only if they both use the same authentication type (null, clear text, or message digest 5 [MD5]). If they use authentication, the authentication data (password or hash value) also needs to match.

Role of LSA Type 3 in OSPF Interarea Routing




Router B, which is an ABR, executes the SPF algorithm using the area 1 database to compute the best path for each subnet that is available in area 1. Based on this computation, Router B will generate a type 3 LSA, which is then injected into the area 0 database. These type 3 LSAs include the cost that Router B has computed for each of these prefixes. Any other router in area 0, such as Router C or D, executes its own SPF algorithm based on the area 0 database. They then add the cost in the type 3 LSAs to their computed cost to Router B to find their total cost to the prefixes in area 1. After Router D has computed these costs, it generates type 3 LSAs for the prefixes from area 1 and injects these into the area 2 database. Any router in area 2, such as Router E, can now compute its own best path to Router D and add this cost to the cost advertised by Router D in the type 3 LSAs to find the total path cost to each prefix in area 1.

Note:

You can configure multiple OSPF process per router and to configure certain interfaces to participate in certain process using the following command:

```
ip ospf process-number area area-number
```

This alternative was first created to be used for unnumbered interfaces. It is interesting to note that in IPv6, this is the normal way of activating an OSPFv3 process on an interface.

**Tshoot Commands:**

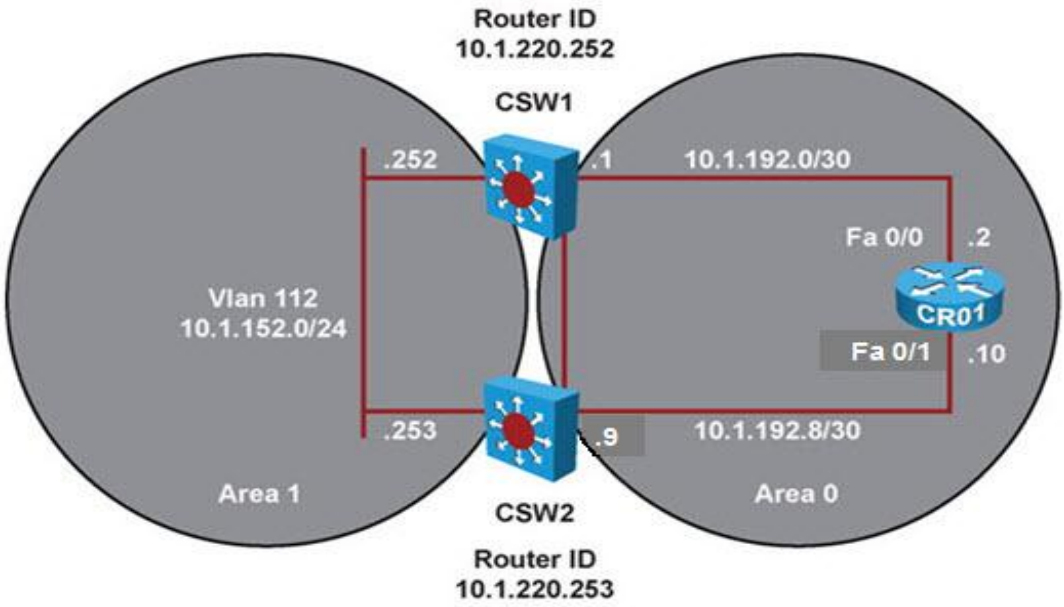
```
show ip ospf interface brief
show ip ospf neighbor
show ip ospf database
show ip ospf statistics
```

This command shows how often and when the SPF algorithm was last executed. This command can be helpful when diagnosing routing instability.

```
debug ip routing
debug ip ospf packet
debug ip ospf events
debug ip ospf adj
debug ip ospf monitor
```

This command monitors when the SPF algorithm is scheduled to run and displays the triggering LSA and a summary of the results after the SPF algorithm has completed.

**Troubleshooting Example: Routing Problem in an OSPF Network**




When you examine the routing table on router CRO1, you only find a single entry, the path through router CSW1.

```
CR01#show ip route 10.1.152.0/24
Routing entry for 10.1.152.0/24
     Known via "ospf 100", distance 110, metric 2, type inter area
     Last update from 10.1.192.1 on FastEthernet0/0, 00:00:11 ago
     Routing Descriptor Blocks:
     * 10.1.192.1, from 10.1.220.252, 00:00:11 ago, via FastEthernet0/0
           Route metric is 2, traffic share count is 1
```

This result is unexpected because CR01 shows that two equal-cost paths are available to CRO1, one through CSW1 and one through CSW2.

- First make sure that you can reach CSW2 by issuing the ping command to the SVI interface of each.

- if you can reach them so, the problem is in a higher layer, There are two main reasons why this could be happening. Either router CSW2 is not advertising subnet 10.1.152.0/24 to area 0, or it might be advertising the route, but the cost to reach subnet10.1.152.0/24 through router CSW2 from router CRO1 is considered to be worse than the cost through router CSW1. To find out whether router CSW2 is advertising subnet 10.1.152.0/24 to area 0, you can consult the OSPF database in router CRO1. It is expected that both routers CSW1 and CSW2 advertise a type3 summary LSA for subnet 10.1.152.0/24.

```
CRO1# show ip ospf database summary 10.1.152.0
               OSPF Router with ID (10.1.220.1) (Process ID 100)
               Summary Net Link States (Area 0)
Routing Bit Set on this LSA
   LS age: 201
   Options: (No TOS-capability, DC, Upward)
   LS Type: Summary Links(Network)
   Link State ID: 10.1.152.0 (summary Network Number)
   Advertising Router: 10.1.220.252
   LS Seq Number: 80000001
   Checksum: 0x1C97
   Length: 28
   Network Mask: /24 TOS: 0 Metric: 1
   LS age: 136
   Options: (No TOS-capability, DC, Upward)
   LS Type: Summary Links(Network)
   Link State ID: 10.1.152.0 (summary Network Number)
   Advertising Router: 10.1.220.253
   LS Seq Number: 80000001
   Checksum: 0x169C
   Length: 28
   Network Mask: /24 TOS: 0 Metric: 1
```

- To verify whether router CRO1 has established a proper neighbor relationship with router CSW1.

```
CRO1# show ip ospf neighbor
Neighbor ID   Pri State     Dead Time   Address      Interface
10.1.220.252   1   FULL/DR  00:00:33    10.1.192.1   FastEthernet0/0

CRO1# show ip ospf interface brief
Interface   PID   Area    IP Address/Mask   Cost   State Nbrs F/C
Lo0         100   0       10.1.220.1/32       1    LOOP  0/0
Fa0/0       100   0       10.1.192.2/30       1    BDR   1/1
```

So CR01 does not advertise ospf routers over f0/1, next you need to see if it is passive for ospf or the network parameters are misconfigured:

```
CRO1# show running-config | section router ospf
router ospf 100
 log-adjacency-changes
 network 10.1.192.2 0.0.0.0 area 0
 network 10.1.192.9 0.0.0.0 area 0
 network 10.1.220.1 0.0.0.0 area 0
```

It is obvious that a problem exists with one of the network statements. The statement `network 10.1.192.9 0.0.0.0 area 0` matches IP address10.1.192.9, which is not one of router CRO1's IP addresses, but an IP address of router CSW2. This is clearly a configuration mistake, and the network statement needs to be replaced with the statement `network 10.1.192.10 0.0.0.0 area`

0 or some other network statement that matches IP address 10.1.192.10 (which is the IP address of router CRO1 on interface F0/1.

## Troubleshooting Route Redistribution

Older routing protocols, such as RIP do not have the capability to mark routes as external in their routing update messages. Newer routing protocols, such as EIGRP, OSPF, and IS-IS, can mark routes as external and prefer internal routes over external routes. This is an important mechanism in the prevention of routing loops caused by route feedback.

Note that the redistribution process is the process that takes the routes from the routing table, not the process that installs the routes in the routing table. Therefore, redistribution is always configured under the "destination" protocol for the routing information. For example, when OSPF routes are redistributed into EIGRP, this is configured under the EIGRP process.

## Seed Metric:

When the route is taken from the routing table and imported into the other protocol's data structures, a metric for the redistributing protocol needs to be attached to that route. This metric is not computed from the metric of the original protocol through the use of some formula. A starting metric, or seed metric, should be configured, which will then be attached to all redistributed routes by the router. **If no seed metric is configured, a default value for the redistributing protocol is used. For some protocols, such as RIP and EIGRP, the default metric is the maximum possible value, which represents "infinity" or "unreachable."** This is important to know when troubleshooting redistribution issues because redistribution into these protocols will fail without explicit configuration of a seed metric. The seed metric can be configured using the default-metric command or along with each redistribution statement (as a parameter or within a route map).

Two important conditions must be met for a prefix learned from one protocol (using redistribution) to be successfully advertised through another protocol:

• **The route needs to be installed in the routing table:** The route needs to be selected as the best route by the source protocol and, if routes from competing sources are present, the route will need to have a lower administrative distance than the competing routes.

• **A proper seed metric is assigned to the redistributed route:** The route needs to be redistributed in the destination protocol data structures with a valid metric for the destination protocol.

## Verifying and Troubleshooting Route Propagation:

When troubleshoot the redistribution process make sure of the following:

• **Troubleshooting the source routing protocol:** Routes can be redistributed only if they are present in the routing table of the redistributing router. If routes are not redistributed as expected, you first need to confirm that they are learned on the redistributing router via the source protocol. Next, you have to check that the route is installed in the routing table.

• **Troubleshooting route selection and installation:** To redistribute a route, it needs to be selected and successfully installed in the routing table by the source protocol. When routes are redistributed between routing protocols in two directions (to and from one protocol to another), it is possible that routing information that originates in one of the routing domains is redistributed into the other routing domain and eventually propagates back to a router that is also connected to the source domain. This happens when there is a topological loop in the network scheme. If the route is subsequently accepted as a better route than the original source route, suboptimal routing can happen. However, if that route is subsequently redistributed back into the source protocol, it might cause routing loops and routing instability. After diagnosing a suboptimal routing problem or routing loop, changing the administrative distance or filtering routes to influence the route selection and installation process can often solve the problem.

• **Troubleshooting the redistribution process:** If routes are learned and installed in the routing table of the redistributing router, but not inserted into the data structures and advertisements of the redistributing protocol, you should verify the configuration of the redistribution process. **Bad seed metrics**, **route filtering**, or **misconfigured routing protocol process** or **autonomous system numbers** are common causes for the redistribution process to fail.

• **Troubleshooting the destination routing protocol:** After the routing information is inserted into the destination protocol's data structures, the routing information is propagated using that protocol's routing update mechanisms. If the routing information is not properly distributed to all routers in the destination routing domain, you should troubleshoot the routing exchange mechanisms for the destination protocol. Each routing protocol has its own methods of exchanging routing information, including external routing information. Research the specific protocol that you are working with to find out if external routes are handled differently than internal routes. For example, OSPF external routes do not propagate into stub areas.

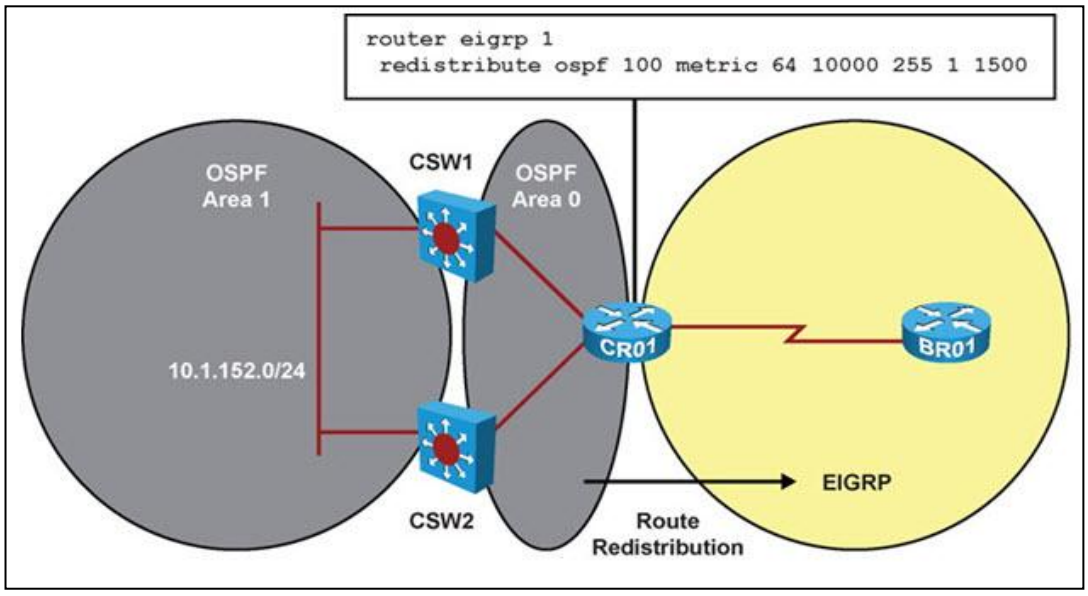
Route Profile Feature:

This feature shows the frequency of routing table changes. First enable this feature using the command: `ip route profile.`

```
R1# show ip route profile
-----------------------------------------------------------------------
Change/     Fwd-path     Prefix     Nexthop     Pathcount     Prefix
interval    change       add        change      change       refresh
-----------------------------------------------------------------------
0           87           87         89          89           89
1           0            0          0           0            0
2           0            0          0           0            0
3           0            0          0           0            0
4           0            0          0           0            0
5           0            0          0           0            0
10          0            0          0           0            0
15          0            0          0           0            0
20          2            2          0           0            0
25          0            0          0           0            0
<output omitted>
```

When the network is stable, only the counters in the first row should increase, because this row represents the number of intervals during which no changes to the routing occurred. When rows other than the first row increase in a situation that you thought to be stable, this could indicate a routing loop.

**Troubleshooting Example: Redistribution from OSPF to EIGRP**




```
CRO1# show ip ospf database | begin Summary
                Summary Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
10.1.152.0      10.1.220.252    472         0x8000003B 0x00A7D1
10.1.152.0      10.1.220.253    558         0x8000003B 0x00A1D6
```

OSPF executes the SPF algorithm and calculates the cost of the path to 10.1.152.0 through switches CSW1 and CSW2. After computing the results, the lowest-cost path is selected for installation into the routing table if no competing routes with a lower administrative distance are present.

```
CRO1# show ip route 10.1.152.0 255.255.255.0
  Routing entry for 10.1.152.0/24
     Known via "ospf 100", distance 110, metric 2, type inter area
     Redistributing via eigrp 1
     Advertised by eigrp 1 metric 64 10000 255 1 1500
     Last update from 10.1.192.9 on FastEthernet0/1, 00:28:24 ago
     Routing Descriptor Blocks:
       10.1.192.9, from 10.1.220.253, 00:28:24 ago, via FastEthernet0/1
          Route metric is 2, traffic share count is 1
     * 10.1.192.1, from 10.1.220.252, 00:28:24 ago, via FastEthernet0/0
          Route metric is 2, traffic share count is 1
```

Both paths through switch CSW1 and switch CSW2 have been installed in the routing table because their costs are identical. The routing table also shows that this route has been marked for redistribution by EIGRP, and the configured EIGRP seed metric is also listed.

```
CRO1# show ip eigrp topology 10.1.152.0 255.255.255.0
IP-EIGRP (AS 1): Topology entry for 10.1.152.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is   42560000
  Routing Descriptor Blocks:
  10.1.192.9, from Redistributed, Send flag is 0x0
    Composite metric is (42560000/0), Route is External
    Vector metric:
      Minimum bandwidth is 64 Kbit
      Total delay is 100000 microseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 0
    External data:
      Originating router is 10.1.220.1 (this system)
      AS number of route is 100
      External protocol is OSPF, external metric is 2
      Administrator tag is 0 (0x00000000)
```

Here, you can clearly see that the route was taken from the routing table and inserted into the topology table as an external route. The five components of the configured seed metric are listed. In addition, some extra parameters were attached to the route to mark that the route was originated by the OSPF protocol with process number 100 and was injected into EIGRP by the router with EIGRP router ID 10.1.220.1 (which is the local router, CRO1).

```
BRO1# show ip eigrp topology 10.1.152.0 255.255.255.0
IP-EIGRP (AS 1): Topology entry for 10.1.152.0/24
   State is Passive, Query origin flag is 1, 1 Successor(s), FD is   43072000
   Routing Descriptor Blocks:
   10.1.193.1 (Serial0/0/1), from 10.1.193.1, Send flag is 0x0
     Composite metric is (43072000/42560000), Route is External
     Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 120000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
     External data:
        Originating router is 10.1.220.1
        AS number of route is 100
        External protocol is OSPF, external metric is 2
        Administrator tag is 0 (0x00000000)


BRO1# show ip route 10.1.152.0 255.255.255.0
Routing entry for 10.1.152.0/24
  Known via "eigrp 1", distance 170, metric 43072000, type external
  Redistributing via eigrp 1
  Last update from 10.1.193.1 on Serial0/0/1, 00:00:35 ago
  Routing Descriptor Blocks:
  * 10.1.193.1, from 10.1.193.1, 00:00:35 ago, via Serial0/0/1
    Route metric is 43072000, traffic share count is 1
    Total delay is 120000 microseconds, minimum bandwidth is 64 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 3/255, Hops 1
```

**Troubleshooting BGP:**

Note the following about BGP operations:
1- BGP neighbors need not be directly connected. Neighbors are manually configured, not automatically discovered through a hello protocol.so neighbors sometimes are called "Peers". A TCP session is established between the predefined neighbors to exchange routing information, and this TCP session can span multiple router hops if necessary. A BGP router attempts to establish a TCP session to the neighbor IP address on TCP port 179. Alternatively, it will accept incoming TCP sessions to port 179, as long as the source IP address matches one of its configured neighbor IP addresses.

2- BGP has 2 tables: The first one is a neighbor table to keep track of the state of configured neighbors. The second is BGP's main data structure, the BGP table, which BGP uses to store all the prefixes, including those received from the neighbors. This table, sometimes called the BGP Routing Information Base (RIB), stores all the locally injected routes, plus all routes that were received from all the router's peers, together with all the BGP attributes that are associated with each route, such as the next hop, autonomous system path, local preference, origin, multi-exit discriminator (MED) or metric, origin code, and community attributes. For each prefix, the BGP best path selection algorithm assesses the usability of the available paths and, if one or more usable paths exist, selects one of the paths as the best path. The best path is subsequently advertised to other BGP peers, and BGP attempts to install this route in the routing table (offers it to the IP routing table). One of the tests that the BGP best path selection algorithm performs is validating the reachability of the next-hop attribute of a path. If the next-hop IP address is reachable as per the IP routing table's content, the path may be considered as a best path candidate. If no match can be found in the routing table for the next-hop address, the path is not considered usable, and it will not be considered as a best path candidate. By default, BGP will select only one best path for each prefix. The BGP Multipath feature allows additional paths to be installed in the IP routing table.

3- Routes learned from neighbors are placed in the BGP table and can be advertised out to other BGP neighbors. Routes learned from internal (IBGP) neighbors are subject to the synchronization rule[1], unless synchronization is off. There are two methods to inject prefixes into the BGP table and advertise them to BGP neighbors:

• The prefixes must be specifically configured under the BGP routing process (using the network statement).

• The prefixes must be redistributed into BGP (from connected, static, or another interior routing protocol).

[1] This rule states that BGP and IGP must be synchronized before networks learned from an IBGP neighbor can be used, this method results in a very big routing table in IGP which are not designed for this purpose. So you can instead use IBGP on all routers on the transit path.

In both cases, a prefix needs to be present in the IP routing table before it can be advertised to BGP neighbors.

4- On Cisco routers, BGP routes have an administrative distance of 200, unless they are learned from external BGP neighbors (those with different autonomous system numbers); in which case, the administrative distance is 20.

5- Best Paths that are selected as best in the BGP table can be advertised to other BGP routers. Several rules, such as the one commonly referred to as IBGP split-horizon rule, govern the advertisement of BGP routes to neighbors. Also, access lists, prefix lists, and route maps may be applied to filter and manipulate the prefixes and their attributes before exchanging them with a neighbor.

6- After a TCP session has been successfully established with the neighbor, the two routers send BGP OPEN messages to exchange basic parameters and capabilities, such as autonomous system number, router ID, hold time, and supported address families. During this phase, each router compares the neighbor's claimed autonomous system number to the autonomous system number its administrator has entered for the neighbor. If these numbers do not match, the session is reset, and the relation is not established. As a result of this process, the following issues are common causes for failure of BGP peering establishment:

• There is no IP connectivity between the local BGP router and the configured peer's IP addresses. Because BGP peers are not necessarily directly connected, both routers need to have an IP path to the configured neighbor IP address in their routing table.

• The source IP address used by the router that initiates the session does not match the configured neighbor IP address on the receiving router.

• The autonomous system number of a BGP router (specified in its HELLO or OPEN message) does not match the autonomous system number its neighbor has configured for it (and expects from it).

After the TCP session has been established and OPEN messages have been successfully exchange and accepted, BGP starts exchanging update messages over the established TCP session. BGP uses an incremental update process. When a BGP peer relationship is first established, each of the peers advertises its best route for each prefix to its peer. Both peers subsequently install the received routes in the BGP table and execute the BGP best path selection algorithm to select the best path for each prefix based on the newly received information. Each time a new best path for a prefix is selected in the BGP table, an update for that prefix is sent to all relevant peers. When a prefix is removed from the BGP table, a WITHDRAW message for the prefix is sent to all relevant peers. To determine which peers need to be updated, the router applies rules such as the IBGP split-horizon rule or any administratively configured filters. BGP updates are commonly filtered or manipulated by use of access lists, prefix lists, or route maps. These tools can either be applied at

the time updates are received, before the prefixes are installed in the BGP table, or at the time updates are transmitted to a peer.

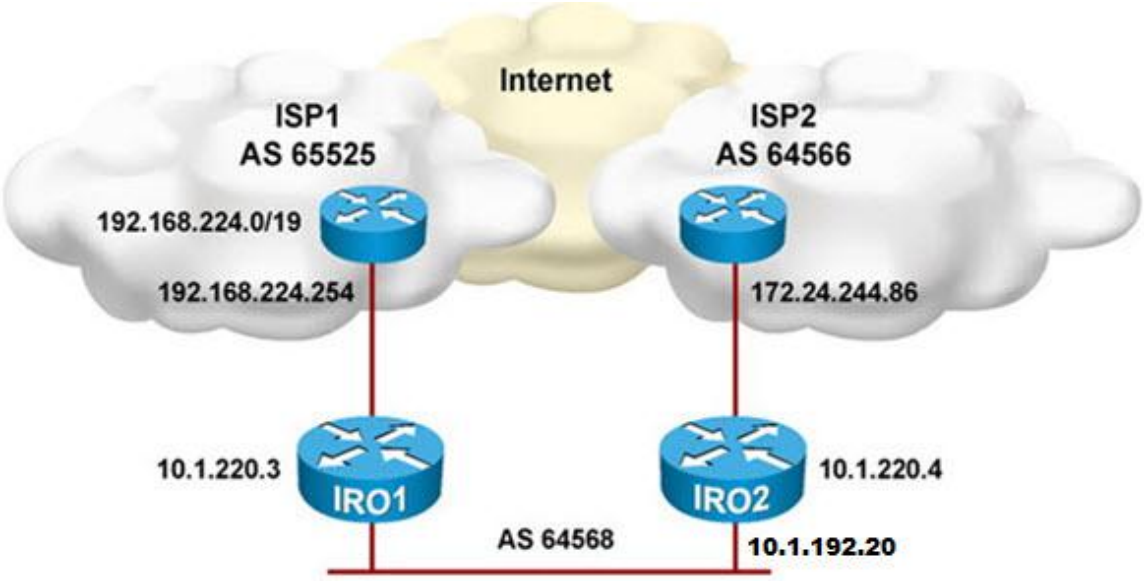
7- Timers: The default timers are 60 (hello) and 180 (hold timer), timers do not need to match to form a BGP neighborship, and the lower of the proposed timers will be used.

## Cisco IOS BGP Commands

```
show ip bgp summary
show ip bgp neighbors
show ip bgp
```
This command displays the content of the BGP table.

```
debug ip bgp:
debug ip bgp updates
debug ip bgp ip-address updates access-list
```
You can use an access list to limit the debug scope.

## Troubleshooting Example: Routing Problem in a BGP Network




The enterprise network shown uses BGP to exchange routing information with two different Internet service providers(ISPs). When a **traceroute** command is executed on router IRO1 to determine the path that is used to reach IP address 192.168.224.1, which belongs to an IP address block that is owned by ISP1, it turns out that the traffic path goes through router IRO2

```
IRO1# trace 192.168.224.1
Type escape sequence to abort.
Tracing the route to 192.168.224.1
  1 10.1.192.20 4 msec 0 msec 0 msec
  2 172.24.244.86 [AS 64566] 4 msec 0 msec 4 msec
  3 192.168.100.1 [AS 65486] 0 msec 4 msec 0 msec
  4 192.168.224.1 [AS 65525] 0 msec * 0 msec
```

1- first check the L3 connectivity between IRO1 and ISP1:

```
IRO1# ping 192.168.224.254
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.224.254, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

2- this is a routing problem. Because BGP is used for external routing, investigation of the BGP operation is in order. There are two likely reasons why the traffic is routed through router IRO2 instead of directly to Internet service provider 1:

• Router IRO1 has not learned about the prefix directly from ISP1 at all.

• Router IRO1 router has learned about the route, but it incorrectly prefers the path through router IRO2.

3- To start the investigation, a good first step is to confirm that the path used to route the traffic to192.168.224.1 is indeed the BGP route learned from router IRO2 and not a route obtained from some other source.

```
IRO1# show ip route 192.168.224.1
Routing entry for 192.168.224.0/19, supernet
Known via "bgp 64568", distance 200, metric 0
Tag 64566, type internal
Redistributing via eigrp 1
Last update from 172.24.244.86 00:24:22 ago
Routing Descriptor Blocks:
* 172.24.244.86, from 10.1.220.4, 00:24:22 ago
Route metric is 0, traffic share count is 1
AS Hops 2
Route tag 64566
```

Shows that the route to network 192.168.224.0/19 is the best match for destination IP address 192.168.224.1 and that this route is an internal BGP path and its source is the router with IP address 10.1.220.4, which is IRO2.

However, the IP routing table does not show whether the path through IRO2 is used because no other path is available or simply because the path through IRO2 was selected as the best path. This information can be obtained from the BGP table. The output of the following show command shows that only the path through router IRO2 is present in the BGP table and no other BGP-learned paths are available.

```
IRO1# show ip bgp 192.168.224.1
BGP routing table entry for 192.168.224.0/19, version 12
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  64566 65525
    172.24.244.86 (metric 30720) from 10.1.220.4 (10.1.220.4)
       Origin IGP, metric 0, localpref 100, valid, internal, best
```

4- The fact that the path through Internet service provider 1 is not present in the BGP table can have several different causes. ISP1 might not advertise the route, or ISP1 advertises it, but router

IRO1 rejects or ignores the advertisement, or possibly router IRO1 and ISP1 have not successfully established a peering relationship and no routes have been exchanged at all. At this point, a good next step is to start with the third option and investigate whether a neighbor relationship has been established between routers IRO1 and the ISP1 router.

```
IRO1# show ip bgp summary
```

| Neighbor | V | AS | MsgRcvd | MsgSent | TblVer | InQ | OutQ | Up/Down | State/PfxRcd |
|---|---|---|---|---|---|---|---|---|---|
| 10.1.220.4 | 4 | 64568 | 82 | 80 | 14 | 0 | 0 | 01:12:02 | 6 |
| 192.168.224.244 | 4 | 65525 | 0 | 0 | 0 | 0 | 0 | never | Active |

The output reveals that the peering to IP address 10.1.220.4 (IRO2) has been established and six prefixes have been received from the neighbor, while the peering to IP address192.168.224.244 is in the Active state. This means that this router is trying to establish a TCP session to neighbor 192.168.224.244, but has not succeeded yet.

The 192.168.224.244 peer is not the correct IP address and IP address192.168.224.254 should have been used instead for the peering to ISP1 instead.

```
IRO1(config)# router bgp 64568
IRO1(config-router)# no neighbor 192.168.224.244
IRO1(config-router)# neighbor 192.168.224.254 remote-as 65525
IRO1(config-router)# end

IRO1# show ip bgp summary | begin Neighbor
Neighbor        V    AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down  State/PfxRcd
10.1.220.4      4 64568     146     146     19   0    0 02:15:17          5
192.168.224.254 4 65525      14      12     19   0    0 00:03:23          5

IRO1# show ip bgp 192.168.224.0
BGP routing table entry for 192.168.224.0/19, version 17
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Advertised to update-groups:
2
65525
192.168.224.254 from 192.168.224.254 (192.168.100.1)
Origin IGP, metric 0, localpref 100, valid, external, best
```

# Chapter 6. Troubleshooting Addressing Services

**Troubleshooting Common NAT/PAT Issues:**

Global addresses are not usually applied to the inside physical network segments, but they must be advertised to the outside world so that outsiders know where and how to send (respond) packets back. Another consideration must be given to the size of the NAT pools because they have a critical effect on specific troubleshooting scenarios where dynamic NAT is involved. Finally, during every stage of the troubleshooting process, you must consider configuration errors of all sorts.

Consider the following when configuring NAT:

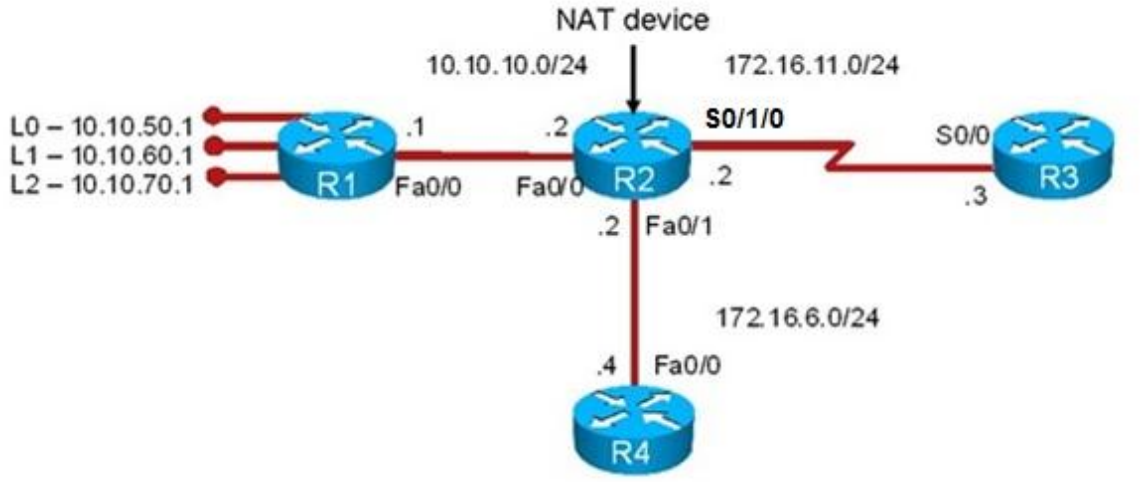
- ACLs are used to tell the NAT device "what source IP addresses are to be translated," and IP NAT pools are used to specify "to what those addresses translate," as packets go from IP NAT inside to IP NAT outside.

- Marking the IP NAT inside interfaces and the IP NAT outside interfaces correctly is very important; otherwise, NAT could have unpredictable and undesirable effects.

- NAT packets still have to obey routing protocols and reachability rules, so make sure that every router knows how to reach the desired destinations. Make sure the public addresses to which addresses translated, are advertised to the outside neighbors and autonomous systems.

```
clear ip nat translation
show ip nat translations
show ip nat statistics
debug ip nat
debug ip packet access-list
debug condition interface interface
```

**Troubleshooting Example: NAT/PAT Problem Caused by a Routing Issue**




In this case, router R1 can ping R4, but router R1 cannot ping R3. You do not have much more information, except that there are no routing protocols running in any of the routers and R1 uses R2 as its gateway of last resort. Your objective is to restore end-to-end connectivity from R1 to all destinations.

```
R2# sh ip nat statistics
Total active translations: 1 (1 static, 0 dynamic, 0 extended)
Outside interfaces:
FastEthernet0/1, Serial0/1/0, Loopback0
Inside interfaces:
FastEthernet0/0
```

So interfaces are configured correctly.

```
R2# sh ip nat translations
Pro Inside global   Inside local   Outside   local Outside global
--- 172.16.6.1      10.10.10.1      ---       ---
```

The only entry in the NAT translation table is the static translation for 10.10.10.1 into 172.16.6.1. The address 10.10.10.1 is the IP address of R1's Fast Ethernet interface (fa0/0). This translation might be causing the problem. Typical issues with static translations occur when there is no route back to the statically translated address, or when the statically selected global address overlaps with an available address in the dynamic address pool.

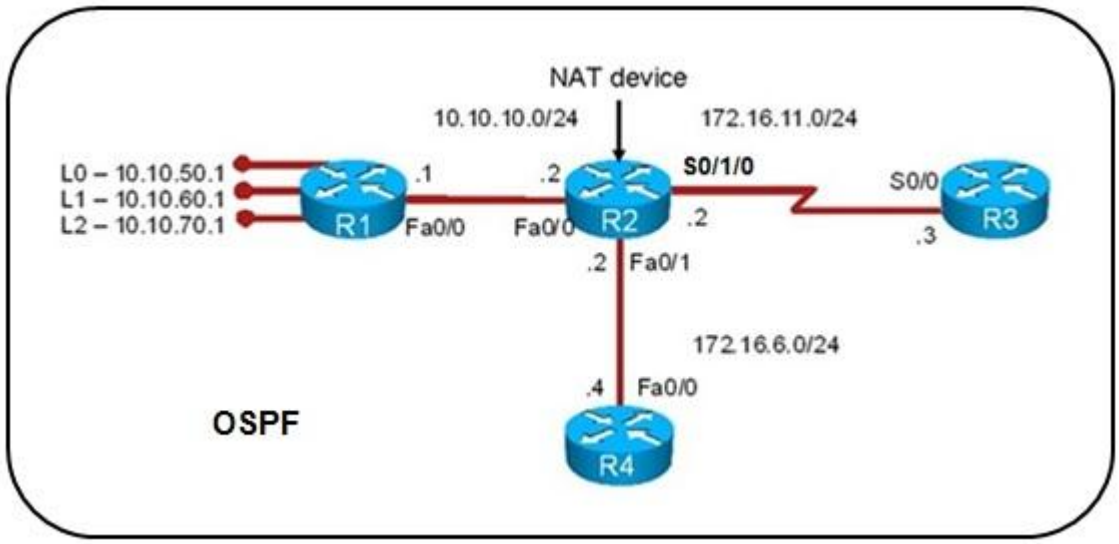
```
R3# show ip route 172.16.6.0
% Subnet not in table
R3#
R3# configure terminal
R3(config)# ip route 172.16.6.0 255.255.255.0 172.16.11.2
R3(config)# exit
```

Now try to ping again:

```
R1# ping 172.16.11.3
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.16.11.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

**Troubleshooting Example: NAT Problem Caused by an Inaccurate Access List**




In this scenario, administrators are reporting that they are unable to use Secure Shell (SSH) from the 10.10.10.0/24 network to routers R3 or R4, but they can accomplish connectivity from the R1 loopbacks. In addition, the risk management team recently performed an upgrade to router and firewall security policies, and some of the changes might have affected the NAT configuration and operations. The routing protocol used is the Open Shortest Path First (OSPF) Protocol, single-area model. Your mission is to restore end-to-end connectivity and make sure SSH is operational to support management processes.

1- First check the L3 connectivity:

```
R1# ping 172.16.11.3 source 10.10.50.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.3, timeout is 2 seconds:
Packet sent with a source address of 10.10.50.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
R1#

R1# ping 172.16.11.3 source 10.10.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.3, timeout is 2 seconds:
Packet sent with a source address of 10.10.10.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
R1#

R1# ssh -l user 172.16.11.3
% Connection refused by remote host
R1#
```

the next logical step is to review possible access lists or security controls. In this instance, however, you will use a clever tool to discover the potential filter: `debug ip tcp transactions`, instead of trying to find ACLs in each router along the path.

```
R1# debug ip tcp transactions
TCP special event debugging is on
R1# ssh -l user 172.16.11.3
% Connection refused by remote host
R1#
*Aug 23 14:59:42.636: TCP: Random local port generated 42115, network 1
*Aug 23 14:59:42.636: TCB63BF854C created
*Aug 23 14:59:42.636: TCB63BF854C bound to UNKNOWN.42115
*Aug 23 14:59:42.636: TCB63BF854C setting property TCP_TOS (11) 62AAF6D55
*Aug 23 14:59:42.636: Reserved port 42115 in Transport Port Agent for TCP IP
type 1
*Aug 23 14:59:42.640: TCP: sending SYN, seq 1491927624, ack 0
*Aug 23 14:59:42.640: TCP0: Connection to 172.16.11.3:22, advertising MSS 536
*Aug 23 14:59:42.640: TCP0: state was CLOSED -> SYNSENT [42115 ->
   172.16.11.3(22)]
*Aug 23 14:59:42.640: TCP0: state was SYNSENT -> CLOSED [42115 ->
   172.16.11.3(22)]
*Aug 23 14:59:42.640: Released port 42115 in Transport Port Agent for TCP IP
   type 1 delay 240000
*Aug 23 14:59:42.640: TCP0: bad seg from 172.16.11.3 — closing connection:
   port 42115 seq 0 ack 1491927625 rcvnxt 0 rcvwnd 0 len 0
*Aug 23 14:59:42.640: TCP0: connection closed - remote sent RST
*Aug 23 14:59:42.640: TCB 0x63BF854C destroyed
```

```
R3# sh ip int s0/1/0
Serial 0/1/0 is up, line protocol is up
Internet address is 172.16.11.3/24
Broadcast address is 255.255.255.255
Address determined by nonvolatile memory
MTU is 1500 bytes
Helper address is not set
Directed broadcat forwarding is disabled
Multicast reserved groups joined: 224.0.0.5
Outgoing access list is not set
Inbound access list is FIREWALL-INBOUND
Proxy ARP is enabled
Local Proxy ARP is disabled
Security level is default
Split horizon is enabled
ICMP redirects are always sent
ICMP unreachables are always sent
ICMP mask replies are never sent
IP fast switching is enabled
IP fast switching on the same interface is enabled
IP Flow switching is disabled
IP CEF switching is enabled
IP CEF Feature Fast switching turbo vector
IP multicast fast switching is enabled

R3# sh access-lists
Standard IP access list 11
10 permit any
Extended IP access list FIREWALL-INBOUND
10 permit tcp any host 172.16.11.3 eq www
```

```
20 permit tcp any host 172.16.11.3 eq telent
30 permit tcp any host 172.16.11.3 eq 22
40 permit tcp any host 172.16.11.3 eq ftp
50 permit tcp any host 172.16.11.3 eq ftp-data
60 permit ospf any any (20 matches)
70 deny ip any any (1 match)
```

The output of show ip int serial 0/1/0 shows that an access list called FIREWALL-INBOUND is applied to serial 0/1/0 interface on the inbound direction. Next, you look at the content of the FIREWALL-INBOUND access list using the show access-lists command, and the access list looks correct: statement number 30 permits TCP connection to 172.16.11.3 TCP port number 22

2- As an attempt to find out why the SSH packets from R1 are rejected by R3, you cautiously make use of **debug ip packet** on R3. You re-attempt the SSH session from R1 to R3 and observe the **debug** output on R3.

```
R1# ssh -l user 172.16.11.3
% Connection refused by remote host


R3# debug ip packet
IP packet debugging is on
R3#
R3#
*Aug 23 16:32:42.711: IP: s=172.16.11.2 (Serial0/1/0), d=224.0.0.5, len 80,
  rcvd 0
*Aug 23 16:32:49.883: %SEC-6-IPACCESSLOGP: list FIREWALL-INBOUND denied tcp
10.10.10.1(29832) -> 172.16.11.3(2222), 1 packet
*Aug 23 16:32:49.883: IP: s=10.10.10.1 (Serial0/1/0), d-172.16.11.3, len 44,
access denied
  *Aug 23 16:32:49.883: IP: tableid=0, s-172.16.11.3 (local), d=10.10.10.1
(Serial0/1/0), routed via FIB
  *Aug 23 16:32:49.883: IP: s=172.16.11.3 (local), d=10.10.10.1 (Serial0/1/0),
len 56, sending
  *Aug 23 16:32:50.067: IP: s=172.16.11.3 (local), d=224.0.0.5 (Serial0/1/0),
```

The SSH attempt from R1 fails again, but the security message (%SEC-6-IPACCESSLOGP) in the output of debug on R3 states that the denied TCP has the source IP address 10.10.10.1 and port number 29832 and destination IP address 172.16.11.3 and port number 2222! The destination port number is 2222 instead of 22, which is the allowed port (SSH) in access list FIREWALL-INBOUND that is applied inbound to R3's serial0/1/0 interface. Now you know why the packet is denied, but you have to find out why the destination port number is 2222 rather than 22. In other words, you have to determine which device has translated the port number from 22 to 2222. Based on the network topology, the prime suspect is NAT on R2. Once again, cautiously, you use debug ip nat on R2 and observe the results as you re-attempt SSH from R1 to R3.

```
R2# debug ip nat
IP NAT debugging is on
R2#
*Aug 23 16:28:31.731: NAT*: TCP s=55587, d=22->2222

R1# ssh -l user 172.16.11.3
% Destination unreachable; gateway or host down
```

```
R2# sh ip nat translations
Pro Inside global     Inside local       Outside local      Outside global
tcp ---               ---                172.16.11.3:22     172.16.11.3:2222
tcp 10.10.10.1:29832  10.10.10.1:29832   172.16.11.3:22     172.16.11.3:2222
tcp 10.10.10.1:43907  10.10.10.1:43907   172.16.11.3:22     172.16.11.3:2222
tcp 10.10.10.1:55587  10.10.10.1:55587   172.16.11.3:22     172.16.11.3:2222
tcp 10.10.10.1:60089  10.10.10.1:60089   172.16.11.3:22     172.16.11.3:2222
tcp 10.10.10.1:62936  10.10.10.1:62936   172.16.11.3:22     172.16.11.3:2222
```

The output clearly shows that R2 is port mapping. It is translating port 22 to port 2222, and that is the problem. It seems that the risk management team updated the security policies, but did not update the access lists for the custom ports being used. You are using TCP 2222; but the access list on R3 is permitting TCP 22. The next step is to correct the FIREWALL-INBOUND on R3 and re-attempt SSH from R1 to R3.

```
R3(config)# ip access-list exten FIREWALL-INBOUND
R3(config-ext-nacl)# permit tcp any ho 172.16.11.3 eq 2222


R1# ssh -l user 172.16.11.3
*Aug 23 16:30:26.608: TCP0: state was SYNSENT -> ESTAB [43884 ->
  172.16.11.3(22)]
*Aug 23 16:30:26.608: TCB63BF854C connected to 172.16.11.3.22
```

In summary, the problem was not the NAT configuration, but a lack of synchronization between the configuration teams: The configuration on R2 is doing port mapping to a custom port (2222), but the access list configuration on R3 did not consider or account for the custom port.

**Common DHCP Troubleshooting Issues**

- Remember that, DHCP uses the following UDP ports: (source port is 67, and the destination port is 68).

- Another issue related to DHCP relay agent is that enabling a router interface with the **ip helper-address** command makes the interface forward UDP broadcasts for six protocols (not just DHCP) to the IP address configured using the **ip helper-address** command. Those protocols are as follows:

• TFTP (port 69)

• DNS (port 53)

• Time Service (port 37)

• NetBIOS Name Service and Datagram Service (ports 137 and 138)

• TACACS (port 49)

• DHCP/BOOTP client and server (ports 67 and 68).

If other protocols do not require this service, forwarding their requests must be disabled manually on all routers using the Cisco IOS `no ip forward-protocol udp "port-number"` global configuration mode command.

- TFTP server IP address "150" is a DHCP option that is typically used by devices such as IP phones to download their configuration files

- For troubleshooting purposes, it is important that you can answer the following questions:

• Where are the DHCP servers and clients located? Are they co-located in the same IP subnet, or do you need to configure relay agents?

• Are DHCP relay agents configured? (They are most likely necessary.)

• What are the DHCP pool sizes? Are they sufficient? (Otherwise you'll run out of addresses.)

• Are there any DHCP option compatibility issues? (Some applications will fail if all necessary options are not supplied.) You must also investigate the following possibilities:

• Are there any ACLs or firewalls filtering UDP port 67 or UDP port 68?

• Are there any active DHCP denial-of-service (DoS) attacks?

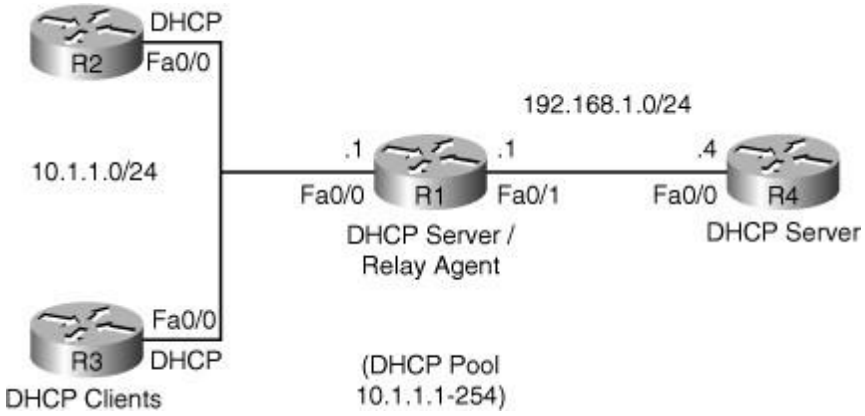
• Is forwarding disabled on the router acting as DHCP Relay Agent for any UDP ports (using the Cisco IOS **no ip forward-protocol udp** *port*command)?

• Is the **ip helper-address** command applied to correct router interfaces?

```
show ip dhcp server statistics
show ip dhcp binding
show ip dhcp conflict
show ip dhcp database
show ip dhcp pool
debug ip udp
debug ip dhcp server [packets | events]
clear ip dhcp binding {* | address}
clear ip dhcp conflict {* | address}
```

## DHCP Troubleshooting Example: Problems After a Security Audit

Router R1 provides DHCP services to the clients in the 10.1.1.0 subnet. The DHCP clients in this example are routers R2 and R3. A security audit has been recently performed in router R1, and you receive reports that R1 is no longer providing reliable DHCP services: The clients are unable to renew their IP addresses.




```
R2# show ip int brief
Interface        IP-Address OK? Method  Status Protocol
FastEthernet0/0  unassigned YES DHCP    up     up

R3# show ip int brief
Interface        IP-Address OK? Method  Status Protocol
FastEthernet0/0  unassigned YES DHCP    up     up
R3# debug dhcp detail
*Aug 23 17:32:48.395: DHCP: Qscan: Timed out Selecting state %Unknown
DHCP problem... No allocation possible


R1# show ip dhcp server statistics
Memory usage         9106
Address pools        1
Database agents      0
Automatic bindings   0
Manual bindings      0
Expired bindings     0
Malformed messages   0
Secure arp entries   0
Message              Received
BOOTREQUEST          0
DHCPDISCOVER         1
DHCPREQUEST          1
DHCPDECLINE          0
DHCPRELEASE          0
DHCPINFORM           0
Message Semt
BOOTREPLY            0
DHCPOFFER            1
DHCPACK              1
DHCPNAK              0
```

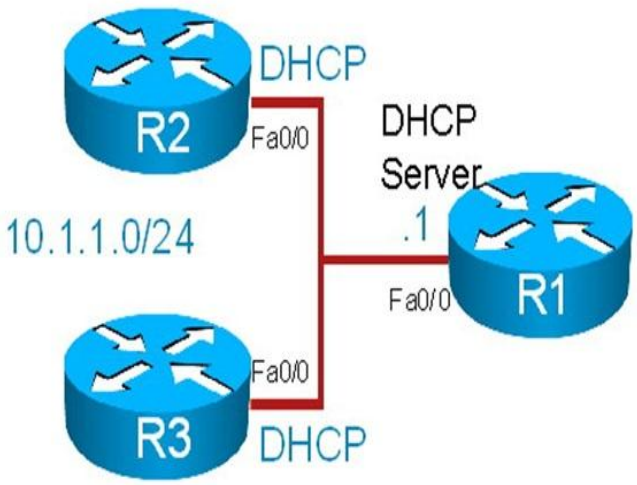
Now you know both the DHCP server and the DHCP clients have the correct configurations, and are operationally UP at physical and data link layers. Remembering that there has been a security audit recently (including one on R1), it is a good idea to verify whether any changes were made by security auditors that affect DHCP. One hardening method that security experts use to make a device less vulnerable to security incidents is shutting down unused services. It is possible that the auditors have shut down the DHCP service on R1. This sounds like a plausible conjecture because the rest of the configuration looks fine. The output of the show ip sockets command displays the active ports on R1, the DHCP server.

```
R1# show ip sockets
Proto Remote Port   Local       Port   In  Out  Stat    TTY OutputIF
88    --listen--    10.1.1.1    10     0   0    0       0
17    --listen--    10.1.1.1    161    0   0    1001    0
17    --listen--    10.1.1.1    162    0   0    1011    0
17    --listen--    10.1.1.1    57767  0   0    1011    0
17    --listen--    --any--     161    0   0   20001    0
17    --listen--    --any--     162    0   0   20011    0
17    --listen--    --any--     60739  0   0    20011   0
```

In the output of the show ip sockets command, you see a series of services functioning on certain open ports; however, DHCP/BOOTP is nowhere to be found. You would need to see UDP 67 if the DHCP service was running. This is certainly a problem, so you would enable the DHCP service using the service dhcp command. After enabling this service, you must wait a few seconds, because DHCP clients retry at different intervals. Meanwhile, you would use the show ip sockets command again and now see port 67 as an active port on R1.

## DHCP Troubleshooting Example: Duplicate Client IP Addresses

The IP address of router R1 on the Fast Ethernet interface was changed from 10.1.1.100 to 10.1.1.1 to comply with the new addressing scheme and policies of the network. This policy states that all branch routers will have the first IP address on any subnet that is being assigned to a network segment. After the change, some DHCP clients are reporting duplicated IP addresses. Clients state that this happens sporadically, a few times a week.




```
R1# show running-config | beg ip dhcp pool
ip dhcp pool vlan10
network 10.1.1.0 255.255.255.0
default-router 10.1.1.1
lease 3

R1# show ip dhcp conflict
IP address  Detection  method   Detection time       VRF
10.1.1.1    Gratuitous ARP      Aug 23 2009 06:28 PM
10.1.1.3    Gratuitous ARP      Aug 23 2009 06:29 PM
10.1.1.3    Gratuitous ARP      Aug 23 2009 06:29 PM
```

**Note:**[2]

- Many devices such as servers and printers are usually configured as DHCP clients and have static IP addresses. If their addresses are not excluded from the DHCP dynamic pool, there will definitely be conflict problems.

```
R1# show running-config | inc excluded
ip dhcp excluded-address 10.1.1.100
```

so the DHCP server does not exclude the static ip addresses assigned to other devices.

```
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# no ip dhcp excluded-address 10.1.1.100
R1(config)# ip dhcp excluded-address 10.1.1.1 10.1.1.20
```

## DHCP Troubleshooting Example: Relay Agent Issue


The DHCP clients in network segment 10.1.1.0 are unable to obtain IP address and other parameters from the central DHCP server. R2 is a DHCP client that is having trouble acquiring the IP address, and R2 is the router that is supposed to act as a relay agent and forward DHCP messages between local clients and the DHCP server (R4).

```
R1# debug ip udp
UDP packet debugging is on
R1#
*Aug 23 19:01:05.303: UDP: rcvd src-0.0.0.0(68), dst=255.255.255.255(67),
length=584
*Aug 23 19:01:05.303: UDP: broadcast packet dropped, src=0.0.0.0,
dst=192.168.1.255
```

---

[2] **A gratuitous ARP** is an ARP broadcast-type of packet that is used to make sure that no other device on the network has the same IP address as the sending device. You should only see gratuitous ARP requests going out; if you see a gratuitous ARP reply, that means another computer on the network has the same your IP address. A gratuitous ARP request is an AddressResolutionProtocol request packet where the source and destination IP are both set to the IP of the machine issuing the packet and the destination MAC is the broadcast address ff:ff:ff:ff:ff:ff. Ordinarily, no reply packet will occur. A gratuitous ARP reply is a reply to which no request has been made. When a machine receives an ARP request containing a source IP that matches its own, then it knows there is an IP conflict.

R1 is certainly receiving DHCP requests. The UDP/IP packets shown have a source address of 0.0.0.0, destination address of 255.255.255.255 with source UDP port of 68 (DHCP client) and destination UDP port of 67 (DHCP server). The problem could be that the fa0/0 interface facing the DHCP client is missing the `ip helper-address` command pointing to 192.168.1.4.

```
R1(config)# int fa0/0
R1(config-if)# ip helper-address 192.168.1.4
R1(config-if)# end
R4#
R4#
R4#debug ip dhcp
*Aug 23 19:31:39.303: UDP: sent src=0.0.0.0(67), dst=255.255.255.255(68),
length=308
*Aug 23 19:31:39.303: UDP: rcvd src=0.0.0.0(68), dst=255.255.255.255(67),
length=584
*Aug 23 19:31:39.303: UDP: sent src=0.0.0.0(67), dst=255.255.255.255(68),
length=308
*Aug 23 19:31:40.159: UDP: rcvd src=0.0.0.0(68), dst=192.168.1.4(67),
length=584
*Aug 23 19:31:44.159: UDP: rcvd src=0.0.0.0(68), dst=192.168.1.4(67),
length=584
```

**Troubleshooting IPv6:**

```
debug ipv6 routing
debug ipv6 nd
```
Neighbor Discovery duplicate addresses, autoconfiguration, and other conditions

```
debug ipv6 packet
show ipv6 interface brief
show ipv6 routers
```
Display IPv6 router advertisement (RA) information received from onlink routers

```
show ipv6 route
show ipv6 protocols
```

**IPv6 Troubleshooting Example: Stateless Autoconfiguration Issue:**


You are informed that recent changes in the network have rendered router R3 isolated and with no connectivity outside the Fast Ethernet segment connected to R1. You verify this by performing a ping from R3 to a remote destination (24::24:2), and it does not succeed. When you inspect the change log you notice that the changes were aimed at providing automatic IP address assignment and configuration to certain devices. After the change, R3 lost connectivity to the rest of the network.

```
1- Beginning from R3, check the L2 connectivity:
R3# show ipv6 interface f0/0
FastEthernet0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::219:55FF: :FEF0:B7D0
Global unicast address(es):
13::13:3, subnet is 13::/64
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF13:3
  FF02::1:FFF0:B7D0
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

Hosts use stateless autoconfig for addresses, it is good to check the configuration of the fa0/0:

```
R3# show run int f0/0
! Interface FastEthernet0/0
    no ip address
    duplex auto
    speed auto
    ipv6 address autoconfig
    ipv6 enable
    end

R3# show ipv6 route
IPv6 Routing Table - 6 entries
!
C     13::/64 [0/0]
        via ::, Fast Ethernet0/0
L     13:219:55FF;FEF0:B7D0/128 [0/0]
        via ::, FastEthernet0/0
C     103::/64 [0/0]
         via ::, Loopback0
L     103::3/128 [0/0]
         via ::, Loopback0
L     FE80::/10 [0/0]
          via ::, Null0
L     FF00::/8 [0/0]
          via ::, Null0
```

At this point, your line of thinking is that R3 has an IP address, but no default route. The router is configured for stateless autoconfiguration on that interface, so it should be obtaining a default router through autoconfiguration. You know that the autoconfiguration process is ruled by ICMPv6 as part of the ND set of features.

The best tool to understand this process is the debug ipv6 nd command, to notice the process of neighbor relationship:

```
R3# debug ipv6 nd
ICMP Neighbor Dicovery events debugging is on
R3#
R3# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)# int f0/0
R3(config-if)# sh
R3(config-if)# no shu
R3(config-if)#
*Aug 23 21:44:18.491: ICMPv6-ND: Sending Final RA on FastEthernet0/0
*Aug 23 21:44:18.491: ICMPv6-ND: Address FE80::219:55FF:FEF0:B7D0/10 is down on
FastEthernet0/0
*Aug 23 21:44:20.491: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
to up
*Aug 23 21:44:20.971: ICMPv6-ND: Sending NS for FE80::219:55FF:FEF0:B7D0 on
FastEthernet0/0
*Aug 23 21:44:21.971: ICMPv6-ND: DAD: FE80::219:55FF:FEF0:B7D0 is unique
*Aug 23 21:44:21.971: ICMPv6-ND: Sending NA for FE80::219:55FF:FEF0:B7D0 on
FastEthernet0/0
*Aug 23 21:44:21.971: ICMPv6-ND: Address FE80::219:55FF:FEF0:B7D0 is up on
FastEthernet0/0
*Aug 23 21:44:23.971: ICMPv6-ND: Sending RS on FastEthernet0/0
*Aug 23 21:44:27.971: ICMPv6-ND: Sending RS on FastEthernet0/0
*Aug 23 21:44:31.971: ICMPv6-ND: Sending RS on FastEthernet0/0
```

At this point, you would shift your troubleshooting effort to R1 and look at the interface connecting to R3 first. Interfaces must be active and have a valid IPv6 address to advertise prefix and autoconfiguration information using ICMPv6.

```
R1# show running-config interface f0/0

! interface FastEthernet0/0
   no ip address
   duplex auto
   speed auto
   ipv6 address 13::13:1/64
   ipv6 enable
end
R1#
R1# show ipv6 interface f0/0
FastEthernet0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::219:56FF:FE2C:9856
Global unicast address(es): 13::13:1, subnet is 13::/64
Joined group address(es):
FF02::1
FF02::2
FF02::1:FF13:1
FF02::1:FF2C:9856
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
Default router is FE80::219:55F:FEF0:B7D0 on FastEthernet0/0
```

On R1, you must verify one of the requirements for autoconfiguration, which is having IPv6 unicast routing explicitly enabled:

```
R1# show run | inc unicast-routing
R1#

R1(config)# ipv6 unicast-routing
```

Notice that R3's fa0/0 interface previously had a valid IPv6 address, but no valid lifetime for the IPv6 address. Now you notice that the valid lifetime and preferred lifetime parameters are present

```
R1# sh ipv6 int f0/0
FastEthernet0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::219:55FF:FEF0:B7D0
Global unicast address(es):
13::219:55FF:FEF0:B7D0, subnet is 13::/64 [PRE]
Valid lifetime 2591941 preferred lifetime 604741


R3# ping 24::24:2
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 24::24:2, timeout is 2 seconds:
!!!!!
Success reate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

**NOTE:**

**How to enable IPv6 statless autoconfiguration:**

R2 will require his IPv6 configuration parameters from R1

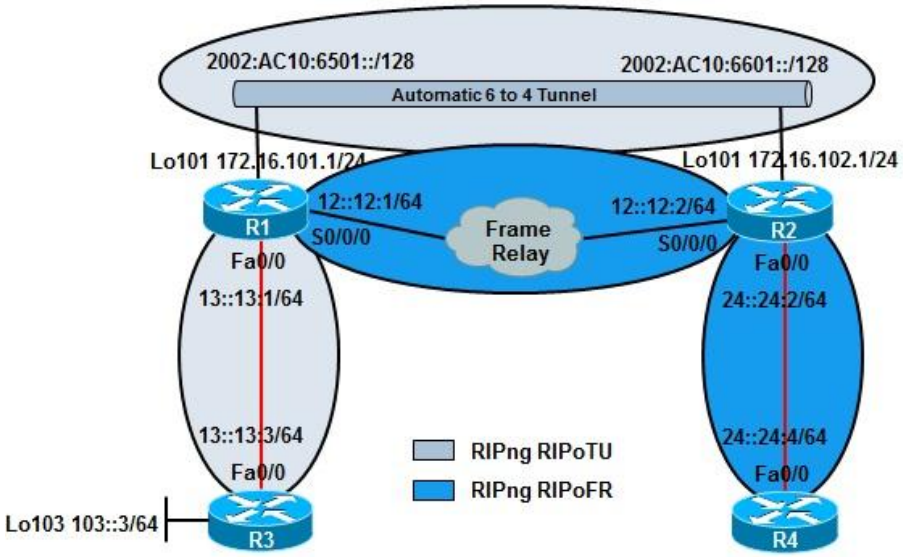

R1:

```
!
ipv6 unicast-routing
ipv6 general-prefix IPV6_TEST 2001:ABCD::/64
!
interface FastEthernet0/0
  ipv6 address 2001:ABCD::/64 eui-64
  ipv6 enable
```

R2:

```
!
ipv6 unicast-routing
!
interface FastEthernet0/0
 ipv6 address autoconfig
 ipv6 enable
```

**IPv6 Troubleshooting Example: Redistribution Issue**




R1 and R2 have two connections: one through the 6to4 tunnel, and a main link over a Frame Relay network. There are two RIPng processes running: one between R3 and R1 and then across the tunnel "RIPoTU", and the second process running between R4 and R2 and across the Frame Relay virtual circuit "RIPoFR",. A recent change ticket performed two-way redistribution between the two RIP processes on R2. However, R4 has lost reachability to R3, specifically the loopback interface on R3.

```
R4# show ipv6 route
IPv6 Routing Table - 6 entries
R 12::/64 [120/2]
    via FE80::219:55FF:FE92:A442, Fast Ethernet0/0
R 103:::/64 [120/7]
     via FE80::219:55FF:FE92:A442, Fast Ethernet0/0

R4#ping 103::3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 103::3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Network 103::/64, which is the loopback of R3, is present in R4's routing table, but a ping from R4 to R3's loopback fails.

```
R2# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip RIPoFR"
  Interfaces:
    FastEthernet0/0
    Serial0/0/0
  Redistribution:
    Redistributing protocol rip RIPoTU with metric 5
IPv6 Routing Protocol is "rip RIPoTU"
    Interfaces:
      Loopback101
      Tunnel0
    Redistribution:
       Redistributing protocol rip RIPoFR with metric 15
```

Notice that redistribution on R2 is bidirectional, from RIPoTU to RIPoFR, and vice versa. Technically, R2 should be redistributing RIPoTU into RIPoFR so that it can advertise R3's Ethernet and loopback network addresses, learned through RIPoTU, to R4 through RIPoFR. Performing redistribution of RIPoFR to RIPoTU is better on R1 so that R3 learns about R4's Ethernet and loopback network addresses.

look at R1's routing table to see whether R4 networks, specifically the 24::/64 network, are present.

```
R1# show ipv6 route
IPv6 Routing Table - 6 entries
S ::/0 [1/0]
    via ::, Tunnel0

C 12::/64 [0/0]
    via ::, Serial0/0/0

L 12::12:1/128 [0/0]
    via ::, Serial0/0/0

C 13::/64 [0/0]
    via ::, FastEthernet0/0

L 13::13:1/128 [0/0]
```

```
    via ::, FastEthernet0/0

R 103::/64 {120/2]
    via FE80::219:55FF:FEF0:B7D0, FastEthernet0/0

LC 2002:AC10:6501::1/128 [0/0]
    Via ::, Tunnel0

L FE80::/10 [0/0]
    via ::, Null0

L FF00::/8 [0/0]
    via ::, Null0
```

As the output reveals, the 24::/64 network is not present in R1's IPv6 routing table.

Assuming that redistribution on R2 is causing the problem, you fix the redistribution metric on R2 and modify the redistribute command to still have a high metric, such as 10, but not 15,

```
R2(config)# ipv6 router rip RIPoTU
R2(config-rtr)# redistribute rip RIPoFR metric 10
```

Now, go back to R1, and check its IPv6 routing table again.

```
R1# show ipv6 rip
RIP process "RIPoTU", port 521, multicast-grop FF02::9, pid 178
      Administrative distance is 120. Maximum paths is 16
      Updates every 30 seconds, expire after 180
      Holddown lasts 0 seconds, garbage collection after 120
      Split horizon is on; poison reverse is off
      Default routes are not generated
      Periodic updates 201, trigger updates 13
  Interfaces:
     Tunnel0
     FastEthernet0/0
  Redistribution:
     None
RIP process "RIPoFR", port 521, multicast-grop FF02::9, pid 179
      Administrative distance is 120. Maximum paths is 16
      Updates every 30 seconds, expire after 180
      Holddown lasts 0 seconds, garbage collection after 120
      Split horizon is on; poison reverse is off
      Default routes are not generated
      Periodic updates 197, trigger updates 6
  Interfaces:
      None
Redistribution:
Redistributing protocol rip RIPoTU with metric 5
```

Strangely, you don't see the Frame Relay network. R1 is not receiving any routes through the Frame Relay network (the serial interface). You should verify that IPv6 RIP is properly configured in router R1.

The RIPoFR process has no interfaces enabled and that is one reason R1 is not receiving routes through the Frame Relay link. On R1, you enable the RIPoFR process on the serial interface,

```
R1(config)# int s0/0/0
R1(config-if)# ipv6 rip RIPoFR enable
```

You can also use the `debug ipv6 routing` command to see the process. Now R1 is receiving R2 routes and also R4 routes, but trying to ping 24::/64 from R3 is not successful

```
R3# ping 24::24:2
```

You know that R1 had a route to R4's Ethernet address (24::/64). You also know that R2 is redistributing the Frame Relay routes into RIPoTU and sending them to R1. R1 receives R4's Ethernet address (24::/64) through RIPoFR and RIPoTU, but it ignores the latter because of its large metric. You need to make sure R1 advertises R4's Ethernet address to R3, so you need to return to R1 and fix that. One way to fix this problem is to redistribute RIPoFR into RIPoTU on R1 so that R1 advertises R4's Ethernet to R3.

```
R1# show ipv6 prot
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip RIPoTU"
   Interfaces:
     Tunnel0
     FastEthernet0/0
    Redistribution:
      None
IPv6 Routing Protocol is "rip RIPoFR"
    Interfaces:
      Serial0/0/0
    Redistribution:
      Redistributing protocol rip RIPoTU with metric 5
```

R1 is redistributing RIPoTU into RIPoFR, which is not really necessary here. On the other hand, R1 is not redistributing RIPoFR into RIPoTU, which is necessary, because you want R4's Ethernet address to be advertised to R3. Fixing this is easy. On R1, you go into the RIPoTU process using the ipv6 router rip RIPoTU command and perform redistribution using the redistribute rip RIPoFR include-connected command,

```
R1(config)# ipv6 router rip RIPoTU
R1(config-rtr)# redistribute rip RIPoFR include-connected metric 5
```

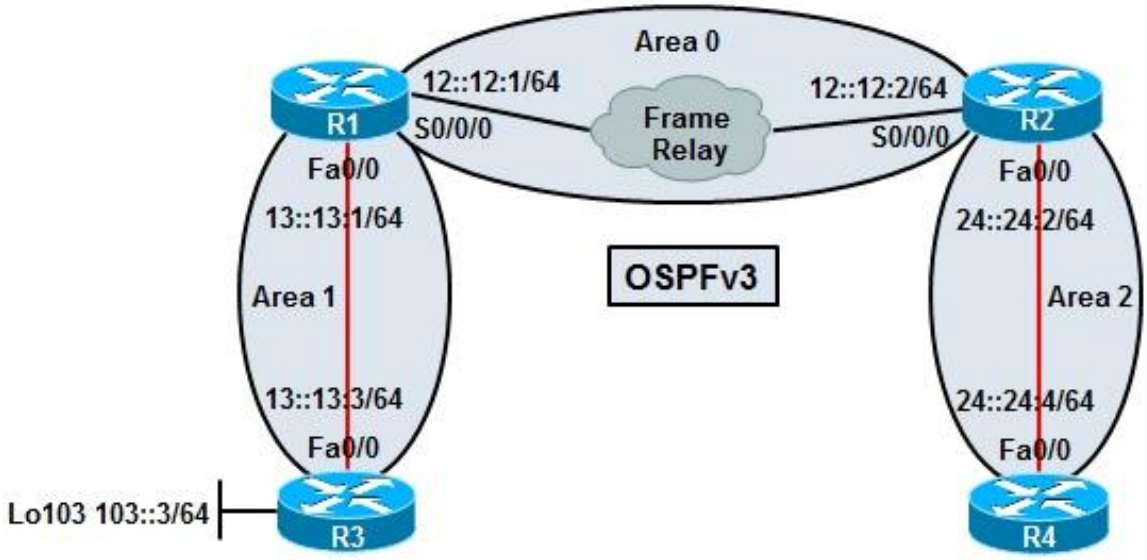
```
R3# ping 24::24:4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 24::24:2, timeout is 2 seconds
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/4 ms
```

**The problem is solved, and it required a lot of effort. There were two main causes:**

**• On R1, IPv6 RIP was not active on its Frame Relay interface.**

**• On R1, the RIPoFR process was not redistributed into the RIPoTU process.**

**IPv6 Troubleshooting Example: OSPFv3 Configuration Errors**




It has been reported that a recent power outage might have caused some of the routers to restart. The network administrators are not completely certain that configurations were properly saved before the incident. End-to-end connectivity has been lost in this network ever since.

```
R4# ping 103::3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 103::3, timeout is 2 seconds:
..
R2# ping 103::3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 103::3, timeout is 2 seconds:
..
R1# ping 103::3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 103::3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1# ping 13::13:3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 13::13:3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
R1#
R2# ping 103::3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 103::3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R2# ping 13::13:3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 13::13:3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R2#

R1# show ipv6 ospf interface
Serial0/0/0 is up, line protocol is up
```

```
  Link Local Address FE80::219:56FF:FE2C:9856, Interface ID 5
  Area 0, Process ID 1, Instance ID 0, Router ID 172.16.101.1
  Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Hello due in 00:00:01
  Index 1/1/2, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
FastEthernet0/0 is up, line protocol is up
  Link Local Address FE80::219:56FF:FE2C:9856, Interface ID 3
  Area 1, Process ID 1, Instance ID 0, Router ID 172.16.101.1
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.101.1, local address FE80::219:56FF:FE2C:9856
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
     Hello due in 00:00:01
  Index 1/1/1, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
R1#
R1# show ipv6 ospf neighbor
R1#


R1# debug ipv6 ospf hello
OSPFv3 hello events debugging is on
R1#
R1#
*Aug 24 13:19:49.751: OSPFv3: Rcv hello from 103.103.103.103 area 1 from
FastEthernet0/0 FE80::219:55FF:FEF0:B7D0 interface ID 3
*Aug 24 13:19:49.751: OSPFv3: Hello from FE80::219:55FF:FEF0:B7D0 with
mismatched Stub/Transit area option bit
*Aug 24 13:19:56.087: OSPFv3: Send hello to FF02::5 area 1 on FastEthernet0/0
from FE80::219:56FF:FE2C:9856 interface ID 3
*Aug 24 13:19:59.751: OSPFv3: Rcv hello from 103.103.103.103 area 1 from
FastEthernet0/0 FE80::219:55FF:FEF0:B7D0 interface ID 3
*Aug 24 13:19:59.751: OSPFv3: Hello from FE80::219:55FF:FEF0:B7D0 with
mismatched Stub/Transit area option bit
```

The mismatched Stub/Transit area option bit that the debug output shows indicates that either R1 is calling the area a stub and R3 does not, or R3 is calling the area a stub and R1 does not. The best way to see the area configuration is to use the show ipv6 ospf command.

```
R1# show ipv6 ospf
  Routing Process "ospfv3 1" with ID 172.16.101.1
  It is an area border router
  SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
  Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
  LSA group pacing timer 240 secs
  Interface flood pacing timer 33 msecs
  Retransmission pacing timer 66 msecs
  Number of external LSA 0. Checksum Sum 0x000000
  Number of areas in this router is 2. 1 normal 1 stub 0 nssa
  Reference bandwidth unit is 100 mbps
    Area BACKBONE(0) (Inactive)
      Number of interfaces in this area is 1
      SPF algorithm executed 3 times
      Number of LSA 4. Checksum Sum 0x01F36B
      Number of DCbitless LSA 0
      Number of indication LSA 0
      Number of DoNotAge LSA 0
     Flood list length 0
Area 1
    Number of interfaces in this area is 1
    It is a stub area, no summary LSA in this area
      Generates stub default route with cost 1
    SPF algorithm executed 5 times
    Number of LSA 4. Checksum Sum 0x016293
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
    Flood list length 0

R3# show ipv6 ospf
Routing Process "ospfv3 1" with ID 103.103.103.103
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Number of external LSA 0. Checksum Sum 0x000000
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Reference bandwidth unit is 100 mbps
   Area 1
     Number of interfaces in this area is 1
     SPF algorithm executed 1 times
     Number of LSA 3. Checksum Sum 0x02286D
     Number of DCbitless LSA 0
     Number of indication LSA 0
     Number of DoNotAge LSA 0
     Flood list length 0
R3#

R3(config)# ipv6 router ospf 1
R3(config-rtr)# area 1 stub
Sep 27 20:17:38.747: %OSPFv3-5-ADJCHG: Process 1, Nbr 172.16.101.1 on
FastEthernet0/0 from LOADING to FULL. Loading Done
R1# ping 103::3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 103::3, timeout is 2 seconds:
!!!!!
```

- Next  Adding the Frame Relay Maps on R2 and R1 for Link-Local Addresses:

```
R1# show frame-relay map
Serial0/0/0 (up): ipv6 12::12:2 dlci 122(0x7A, 0x1CA0), static
             broadcast
             CISCO, status defined, active

R2# show ipv6 interface brief
FastEthernet0/0              [up/up]
    FE80::219:55FF:FE92:A442
    24::24:2
Serial0/0/0                  [up/up]
    FE80::219:55FF:FE92:A442
    11::1:2
    12::12:2
Loopback101                  [up/up]
    FE80::219:55FF:FE92:A442

R1# conf t
Enter configuration commands, one per line. End with CNTL/Z
R1(config)# int s0/0/0
R1(config-if)# fram map ipv6 FE80::219:55FF:FE92:A442 122 broadcast
R1(config-if)# end

R1# show ipv6 interface brief
FastEthernet0/0              [up/up]
    FE80::219:56FF:FE2C:9856
    13::13:1
Serial0/0/0                  [up/up]
    FE80::219:56FF:FE2C:9856
    11::1:1
    12::12:1
Loopback101                  [up/up]
    FE80::219:56FF:FE2C:9856


R2# conf t
Enter configuration commands, one per line. End with CNTL/A.
R2(config)# int s0/0/0
R2(config-if)# fram map ipv6 FE80::219:56FF:Fe2C:9856 221 br
R2(config-if)# end
R2#
*Aug 24 13:55:00.459: %SYS-5-CONFIG_I: Configured from console by console
*Aug 24 13:56:59.175: %OSPFv3-5-ADJCHG: Process 1, Nbr 172.16.101.1 on
Serial0/0/0
from LOADING to FULL, Loading Done
R2#
R2# ping13::13:3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 13::13:3, timeout is 2 seconds:
!!!!!
```

Now you need to fix the problem at the R2-to-R4 segment. You must first determine whether OSPF is configured on the interfaces with the show ipv6 ospf interface command. On R2, OSPFv3 is enabled on the fa0/0 interface in area 2, but R4, which also has OSPFv3 enabled on its fa0/0 interface, has it in area 0. This is an error. In OSPF, interfaces drop all OSPFv3 packets that do not have a matching OSPFv3 area ID in the packet header.

```
R4(config)# int f0/0
R4(config-if)# ipv6 ospf 1 area 2
```

# Chapter 7. Troubleshooting Network Performance Issues

## Common NetFlow Issues

As efficient as NetFlow is in exporting aggregated information to a NetFlow Analyzer server, you might need to tune it so that it does not create performance degradation in the NetFlow-enabled device by consuming too much memory or CPU cycles. You might have to set limits to the number of entries in the cache, or tune the aging timers of NetFlow. Aging timers define how long a flow remains in the cache before being timed out and exported. If the aging timers are too high, the table can remain full continuously.

Following are some of the common NetFlow export culprits:

• A destination IP address has not been configured.

• A source interface has not been configured.

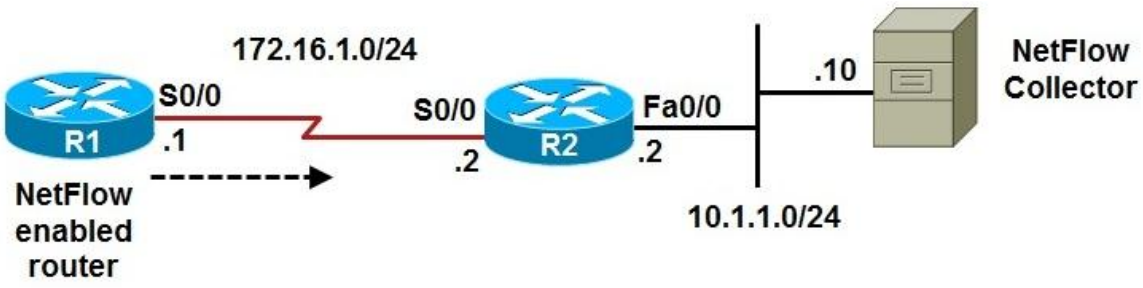
• A source interface has been configured, but does not have an IPv4 address.

• A source interface has been configured, but it is not in the up state.

• The subnet of the destination is unreachable.

```
show ip cache flow
show ip flow export
show ip flow interface
debug ip flow export
```

## NetFlow Troubleshooting Example




The reported problem is that the NetFlow Collector is not receiving data from router R1, one of the NetFlow enabled routers.

1- checking L3 connectivity between R1 and the NetFlow Collector is OK.

2-

```
R1# show ip cache flow
IP packet size distribution (85435 total packets):
! Packet Sizes
  1-32  64  96  128  160  192  224  256  288  320  352  384  416  448  480
 .000 .000 .000 .000 000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

       512  544  576  1024  1536  2048  2560  3072  3584  4096  4608
      .000 .000 .000 .000   1.00  .000  .000  .000  .000  .000  .000

IP Flow Switching Cache, 278544 bytes
  ! Number of Active Flows
  2728 active, 1638 inactive, 85310 added
  463824 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
  last clearing of statistics never
! Rates and Duration
Protocol     Total   Flows   Packets   Bytes  Packets   Active (Sec)  Idle
(Sec)
--------     Flows    /Sec    /Flow     /Pkt   /Sec        /Flow         /Flow
TCP-X            2    0.0        1     1440   11.2         0.0           9.5
TCP-other    82580    11.2       1     1440   11.2         0.0          12.0
Total        82582    11.2       1     1440   11.2         0.0          12.0

! Flow Details Cache
SrcIF       SrcIPaddress   DstIf        DstIPaddress     Pr  SrcP  DstP   Pkts
Et0/0       132.122.25.60  Se0/0        192.168.1.1      06  9AEE  0007      1
Et0/0       139.57.220.28  Se0/0        192.168.1.1      06  708D  0007      1
Et0/0       165.172.153.65 Se0/0        192.168.1.1      06  CB46  0007      1
```

R1 is collecting plenty of data. Next, you check if R1 is exporting the NetFlow data to the correct server using the **show ip flow export** command.

```
R1# show ip flow export
Flow export v5 is enabled for main cache
Exporting flows to 10.1.152.1 (9991)
Exporting using source interface FastEthernet0/0
Version 5 flow records
5 flows exported in 3 udp datagrams
0 flows failed due to lack of export packet
0 export packets were sent up to process level
0 export packets were dropped due to no fib
0 export packets were dropped due to adjacency issues
0 export packets were dropped due to fragmentation failures
0 export packets were dropped due to encapsulation fixup failures
```

the IP address of the NetFlow Collector and the source interface are incorrect. The NetFlow Collector's address is 10.1.152.1 instead of 10.1.1.10, and the source interface is FastEthernet0/0 rather than Loopback0.

3-

```
R1(config)# no ip flow-ex destination 10.1.152.1 9991
R1(config)# ip flow-ex destination 10.1.1.10 9991
R1(config)# no ip flow-export source f0/0
R1(config)# ip flow-export source lo0
```

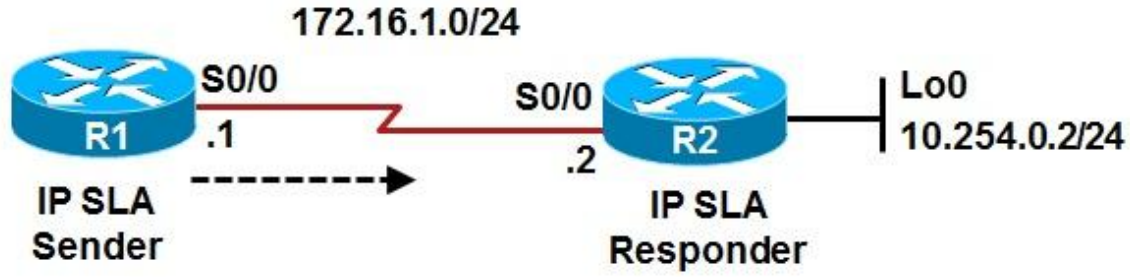
## Common IP SLA Issues

1- Probe scheduling can be problematic if the clock on the device is out of sync; that is the reason synchronizing through Network Time Protocol(NTP) is highly recommended. For example, if you schedule for December 31 at 23:59, but the sender clock is set to last year, the probe will not go out when you expect it to.

2- Network readiness is also essential; when using IP SLAs for troubleshooting purposes, the very same reason that prevents a certain application from working on the network will prevent the probe from working. Typically, it is the firewalls and access control mechanisms that filter or block traffic.

```
show ip sla monitor statistics
show ip sla monitor collection-statistics
show ip sla monitor configuration
debug ip sla monitor trace
```

## IP SLA Troubleshooting Example




In this case, to measure delay, a TCP connection probe (entry 1) is sent on port 2002 from R1 to R2 every 10 minutes, and SNMP traps are sent to an SNMP console if a certain threshold is surpassed. The problem is that the probe does not start and it does not report any statistics.

```
R1# show ip sla monitor configuration
SA Agent, Infrastructure Engine-II
Entry number: 1
Owner:
Tag:
Type of operation to perform: tcpConnect
Target address: 10.254.0.2
Source address: 0.0.0.0
Target port: 2002
Source port: 0
Operation timeout (milliseconds): 60000
Type of service parameters: 0x0
Control packets: enabled
Operation frequency (seconds): 600
Next Scheduled Start Time: 23:59:00
Group Scheduled: FALSE
Life (seconds): Forever
Entry Ageout (seconds): never
Recurring (Starting Everyday): FALSE
Status of entry (SNMP RowStatus): Active
```

```
Threshold (milliseconds): 5000
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 1
Statistic distribution interval (milliseconds): 20
```

The output, displays correct information about probe number 1.

```
R1# show ip sla monitor statistics
Round trip time (RTT) Index 1
Latest RTT: Unknown
Latest operation return code: Unknown
Latest operation start time: Unknown
Number of successes: 0
Number of failures: 0
Operation time to live: Forever
```

```
R1# show run | section ip sla
ip sla monitor 1
type tcpConnect dest-ipaddr 10.254.0.2 dest-port 2002
frequency 600
ip sla monitor schedule 1 life forever start-time 23:59:00 Sep 10
ip sla monitor 2
type echo protocol ipIcmpEcho 10.9.9.21 source-interface FastEhternet0/0
ip sla monitor schedule 2 life forever start-time now
ip sla monitor 3
type udpEcho dest-ipaddr 10.1.1.100 dest-port 5247
ip sla monitor schedule 3 life forever start-time now
```

Using the show run | section ip sla command. Notice that the probe was supposed to start at 23:59, and even though it is past that time, it has not started. Ensure that R1's clock is synchronized with the NTP server.

```
R1# show ntp status
Clock is unsynchronized, stratum 16, no reference clock
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
reference time is CE3D3F49.C3932713 (16:33:13.763 UTC Mon Aug 24 2009)
clock offset is 1.2491 msec, root delay is 22.99 msec
root dispersion is 1.68 msec, peer dispersion is 0.41 msec
```

you notice that it is not synchronized with the NTP server. The phrase no reference clock in the output indicates that the NTP server, which is R2's address 10.254.0.2, is unavailable.

```
R2(config)# ntp master 1
```

```
R1# show ntp status
Clock is synchronized, stratum 2, reference is 10.254.0.2
```

```
R1# sh ip sla monitor statistics
Round trip time (RTT)  Index 1
        Latest RTT: 20 ms
Latest operation start time: 14:31:17.083 UTC Fri Sep 11 2009
Latest operation return code: Ok
Number of successes: 1
Number of failures: 0
Operation time to live: Forever

Round trip time (RTT)  Index 2
        Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: 14:31:41.239 UTC Fri Sep 11 2009
Latest operation return code: No connection
Number of successes: 0
Number of failures: 24
Operation time to live: Forever
```

## Common AutoQoS Issues

Many of the common Cisco AutoQoS issues relate directly to its requirements and limitations, such as not having CEF enabled. Cisco AutoQoS requires CEF to be enabled on the interface, and the interface must be configured with an IP address and a specific (proper) bandwidth.

Note that the bandwidth on a serial interface is not autosensed. Cisco AutoQoS uses the configured interface bandwidth to enable or disable certain QoS features such as compression and fragmentation.

Another common AutoQoS problem cause is mismatched parameters on the two sides of a serial link. If the configured bandwidths differ, for example, Cisco AutoQoS might enable certain features on one side while disabling them on the other side of the same link. This can cause Layer 2 issues and bring the interface down.

Modifying the Cisco AutoQoS configuration after the feature has been enabled can prove problematic, too. For example, if you disable AutoQoS on an interface using the no auto qos command, all Cisco AutoQoS generated commands are removed with the exception of those that have been changed/modified. If you do not remove those commands manually, they might cause errors or performance problems.

Finally, you must know that AutoQoS does not work on an interface if it already has a policy applied to it. Before you apply AutoQoS, confirm that the interface has an IP address, has proper bandwidth configured, has CEF enabled, and has no policies applied to it already.

```
show auto qos interface show auto discovery qos
```

**AutoQoS Troubleshooting Example:**


The connection between routers R1 and R2 is down; however, the service provider maintains that the backbone service is fully operational.

```
R1# show ip interfaces brief
Interface        IP-Address  OK?  Method  Status                  Protocol
FastEthernet0/0  unassigned  YES  unset   up                      up
FastEthernet0/1  unassigned  YES  unset   administratively down  down
Serial0/0/0      172.16.1.1  YES  unset   up                      down
```

serial 0/0/0 is configured for High-Level Data Link Control (HDLC) encapsulation. The documentation, however, indicates that this serial interface must be configured for PPP encapsulation.

```
R1(config)# int s0/0/0
R1(config-if)# encap ppp
R1(config-if)# shut
R1(config-if)# no shut
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to
up
```

Through some investigation and fact gathering, you learn that the problem was reported after someone tried to enable AutoQoS on this interface. Apparently, after encountering errors, that person removed AutoQoS, but the circuit remained down. What that person did not know is that when AutoQoS was removed, the interface encapsulation was changed back to serial interface default encapsulation, which is HDLC. All that was needed was to set the encapsulation to PPP, as done.

However, the issue that remains is that you need to make use of AutoQoS on this interface. AutoQoS (enterprise) has two phases. In the first phase, you need to enable "autodiscovery" on the interface for a while (minimum of 3 days is recommended). In the second phase, you actually enable AutoQoS, which will generate templates and policies based on the discovery enabled in phase one. But notice that the interface went down soon after we entered the auto qos command on serial 0/0/0. It seems that AutoQoS attempted to set the multilink feature on serial 0/0/0 and that failed

```
R1(config)# int s0/0/0
R1(config-if)# auto discovery qos
! Recommended practice dictates letting auto discovery run for some time,
 preferably 3-5 days
R1(config-if)# auto qos
```

AutoQoS attempts the multilink feature for fragmentation purposes, on low bandwidth interfaces. You need to find out what is the configured bandwidth for the serial 0/0/0 interface. R1's running configuration reveals that the bandwidth on serial 0/0/0 is set to 200 (kbps) rather than 2000. This tells you why AutoQoS considered this interface a slow bandwidth interface and attempted the multilink feature on it.

```
R1# sh run int s0/0/0

! interface Serial0/0/0
bandwidth 200
ip address 172.16.1.1 255.255.255.0
ip nbar protocol-discovery
ip flow ingress
encapsulation ppp
auto qos
auto discovery qos
no fair-queue
ppp multilink
ppp multilink group 2001100115
service-policy input TEST
service-policy output TEST
```

The next logical step is to fix the bandwidth configuration on R1's serial 0/0/0 interface and reapply the AutoQoS feature; however, you need to remove AutoQoS first. When you remove the AutoQoS feature, an error occurs and a message indicates that all policies have not been removed. Therefore, when you retry the AutoQoS feature, you receive yet another error message that indicates because there is a policy on the interface, AutoQoS cannot be applied.

```
R1(config)# int s0/0/0
R1(config-if)# no auto qos
% Cannot disable multilink on a multilink group interface
% Not all config may be removed and may reappear after reactivating the
  Logical-interface/sub-interfaces
R1(config-if)# bandwidth 2000
R1(config-if)# auto qos
Policy map TEST is already attached
AutoQoS Error: the following command was not properly applied:
service-policy output AutoQoS-Policy-UnTrust
```

The running configuration shows a service policy called TEST is applied to serial0/0/0 interface for both inbound and outbound traffic. You must remove those lines, reset encapsulation back to PPP, and then reapply AutoQoS. This time AutoQoS succeeds, and the interface stays up. Test the connection between R1 and R2 with a ping to ensure success . It is important to keep in mind that you can only remove policies after verifying they are not necessary. In this case, as the name

implies, the TEST policy was put in place for testing purposes but it was not removed upon test completion

```
R1# ping 172.16.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
```

## Troubleshooting Performance Issues on Switches:

Troubleshooting Switch Interface Performance Problems:

```
show interfaces interface counters
show interfaces interface counters errors
```

This command displays the error statistics for each interface:

**Align-Err:**

This is the number of frames with alignment errors, which are frames that do not end with an even number of octets and have a bad cyclic redundancy check (CRC), received on the port. These usually indicate a physical problem, for example, cabling, a bad port, or a bad network interface card (NIC), but can also indicate a duplex mismatch. When the cable is first connected to the port, some of these errors can occur. Also, if there is a hub connected to the port, collisions between other devices on the hub can cause these errors.

**FCS-Err:**

The numbers of valid size frames with frame check sequence (FCS) errors, but no framing errors. This is typically a physical issue (for example, cabling, a bad port, or a bad NIC), but can also indicate a duplex mismatch.

**Xmit-Err and Rcv-Err:**

This indicates that the internal port transmit (Tx) or receive (Rx) buffers are full. A common cause of Xmit-Err is traffic from a high-bandwidth link that is switched to a lower-bandwidth link, or traffic from multiple inbound links that is switched to a single outbound link. For example, if a large amount of bursty traffic comes in on a Gigabit port and is switched out to a 100-Mbps port, the Xmit-Err field might increment on the 100-Mbps port. This is because the port output buffer is overwhelmed by the excess traffic because of the speed mismatch between the incoming and outgoing bandwidths.

**Undersize:**

The frames received that are smaller than the minimum IEEE802.3 frame size of 64 bytes long, but have a valid CRC.

**Single-Col:** single collision

**Multi-Col**: multi collision

**Late-Col:** late collision.

**Excess-Col:** excessive collision.

**Carri-Sen:**

This occurs every time an Ethernet controller wants to send data on a half-duplex connection. The controller senses the wire and checks whether it is not busy before transmitting. This is normal on a half-duplex Ethernet segment.

**Runts:**

The frames received are smaller than the minimum IEEE 802.3 frame size (64 bytes for Ethernet) and have a bad CRC. This can be caused by a duplex mismatch and physical problems, such as a bad cable, port, or NIC on the attached device.

**Giants:**

These are frames that exceed the maximum IEEE 802.3 frame size (1518 bytes for nonjumbo Ethernet), and have a bad FCS. Try to find the offending device and remove it from the network. In many cases, it is the result of a bad NIC.

```
ASW1# show interfaces FastEthernet 0/1 counters

Port         InOctets       InUcastPkts       InMcastPkts     InBcastPkts
Fa0/1       647140108            499128              4305               0

Port         OutOctets      OutUcastPkts       OutMcastPkts     OutBcastPkts
Fa0/1         28533484            319996                 52                3

ASW1# show interfaces FastEthernet 0/1 counters errors

Port     Align-Err   FCS-Err    Xmit-Err     Rcv-Err   UnderSize   OutDiscards
Fa0/1            0     12618           0       12662           0             0

Port   Single-Col   Multi-Col  Late-Col  Excess-Col  Carri-Sen  Runts   Giants
Fa0/1           0           0         0           0          0      0       44
```

You can see that there are 12618 FCS errors on a total of 499128 + 4305 + 0= 503433 received frames, which translates to 2.5 percent of the received traffic on the interface. In general, more than one FCS error in a million frames (0.0001 percent) is reason to investigate.

- The IEEE 802.3ab Gigabit Ethernet standard mandates the use of autonegotiation for speed and duplex. In addition, although it is not strictly mandatory, practically all Fast Ethernet NICs also use auto negotiation by default.

**Switch Port/Interface Issues:**

No cable connected

• Wrong port

• Device has no power

• Wrong cable type

• Bad cable

• Loose connections

• Patch panels

• Media converters

• Bad or wrong gigabit interface converter (GBIC)

**Auto-MDIX**

Automatic medium-dependent interface crossover (auto-MDIX) is a feature supported on many switches and NICs. This feature automatically detects the required cable connection type (straight-through or crossover) for a connection. As long as one of the two sides of a connection supports auto-MDIX, you can use a crossover or a straight-through Ethernet cable and the connection will work. However, this feature depends on the speed and duplex autonegotiation feature, and disabling speed and duplex negotiation will also disable auto-MDIX for an interface. The default setting for auto-MDIX was changed from disabled to enabled starting from Cisco IOS Software Release 12.2(20)SE.

```
Switch(config)# interface FastEthernet 0/10
Switch(config-if)# shutdown
Switch(config-if)# speed auto
Switch(config-if)# duplex auto
Switch(config-if)# mdix auto
Switch(config-if)# no shutdown
```

```
CSW1# show interface FastEthernet 0/10 transceiver properties
Diagnostic Monitoring is not implemented
Name : Fa0/10
Administrative Speed: auto
Administrative Duplex: auto
Administrative Auto-MDIX: on
Administrative Power Inline: N/A
Operational Speed: 100
Operational Duplex: full
Operational Auto-MDIX: on
Media Type: 10/100BaseTX
```

## The Forwarding Hardware:

the forwarding hardware always consists of two major components:

**Backplane:** internally to a switch a specialized hardware is needed to move frames between ports. This specific part can be called backplane or in some cases we talk of switching fabric. In other words it is the capabilities of the motherboard to switch the traffic using the "buses" which are the circuits on the motherboard. The impact of the backplane on switch performance is limited. The backplane of a switch is designed for very high switching capacity. In most cases, the limiting factor in throughput on a switched network is the capacity of the links between the devices, not the capacity of the backplanes of the switches.

**Decision-making logic:** For each incoming frame, the decision-making logic makes the decision to either forward the frame or discard it; this is also called performing Layer 2 and Layer 3 switching actions. For forwarded frames, the decision-making logic provides the information necessary to rewrite and forward the frame and may take other actions such as the processing of access lists or quality of service (QoS) features.

## Troubleshooting TCAM Problems:

Multilayer switches forward frames and packets at wire speed by using ASIC hardware. Specific Layer 2 and Layer 3 components, such as routing tables or Access Control Lists (ACLs), are **cached into hardware "after being processed by the CPU "software" "**. Routing, switching, ACL and QoS tables are stored in a high-speed table memory so that forwarding decisions and restrictions can be made in high-speed hardware. Switches perform lookups in these tables for result information, such as to determine whether a packet with a specific destination IP address is supposed to be dropped according to an ACL. Cisco Catalyst switches deploys these memory tables using specialized memory architectures, referred to as CAM and TCAM.

This makes TCAM a very important component of Cisco Layer 3 switches and modern routers, since they can store their routing table in the TCAMs, allowing for very fast lookups, which is considerably better than routing tables stored in ordinary RAM.

The decision-making logic of a switch has a significant impact on its performance. The decision-making logic consists of specialized high performance lookup memory, the ternary content-addressable memory (TCAM). The control plane information necessary to make forwarding decisions, such as MAC address tables, routing information, access list information, and QoS information, build the content of the TCAM. The TCAM then takes all the necessary forwarding decisions for a frame at speeds that are high enough and it utilizes full capacity of the switch backplane. TCAM's decision-making process does not impede or limit the forwarding performance of the switch. However, if for some reason frames cannot be forwarded by the

TCAM, they will be handed off (punted) to the CPU for processing. Because the CPU is also used to execute the control plane processes, it can only forward traffic at certain rate. Consequently, if a large amount of traffic is punted to the CPU, the throughput for the traffic concerned will descend, and an adverse effect on the control plane processes will also be observed. TCAM will punt any frames to the CPU for forwarding that it cannot forward itself. This does not include frames that are explicitly dropped (for example, by an access list) because the inbound port is in the spanning-tree Blocking state or because a VLAN is not allowed on a trunk. Traffic might be punted or handled by the CPU for many reasons, some main examples of which are as follows:

• Packets destined for any of the switch IP addresses. Examples of such packets include Telnet, Secure Shell (SSH), or Simple Network Management Protocol (SNMP) packets destined for one of the switch IP addresses.

• Multicasts and broadcasts from control plane protocols such as the Spanning Tree Protocol (STP) or routing protocols. Routing protocol broadcasts and multicasts are processed by the CPU in addition to being flooded to all ports within the VLAN that the frame was received in, as usual.

• Packets that cannot be forwarded by the TCAM because a feature is not supported in hardware. For example, generic routing encapsulation (GRE) tunnels can be configured on a Catalyst 3560 switch, but because this is not a TCAM-supported feature on this switch, the GRE packets will be punted.

• Packets that cannot be forwarded in hardware because the TCAM could not hold the necessary information. The TCAM has a limited capacity, and when entries cannot be programmed into the TCAM, the packets associated to those entries will have to be punted to the CPU to be forwarded. If you have too many IP routes or too many access list entries, some of them might not be installed in the TCAM, and associated packets cannot be forwarded in hardware. This item is the most likely to cause performance problems on a switch. The CPU always handles control plane packets in software, and the volume of this type is relatively low. However, the volume of traffic that flows through a switch is substantial, and if even a fraction of this traffic is handed off to the CPU, it will quickly cause performance degradation. The traffic itself might be dropped or forwarded slowly.

To discover how close the current TCAM utilization is to the platform limits, use the show platform tcam utilization command:

```
CSW1# show platform tcam utilization

CAM Utilization for ASIC# 0                          Max            Used
                                                  Masks/Values   Masks/Values

Unicast mac addresses:                             784/6272         23/99
IPv4 IGMP groups + multicast routes:               144/1152          6/26
IPv4 unicast directly-connected routes:            784/6272         23/99
IPv4 unicast indirectly-connected routes:          272/2176         30/175
IPv4 policy based routing aces:                      0/0            30/175
IPv4 qos aces:                                     768/768         260/260
IPv4 security aces:                               1024/1024         27/27
```

shows the maximum number of masks and values that can be assigned to IP Version 4 not directly connected routes are 272 and 2176, respectively. Currently, 30 masks and 175 values are in use.

For some types of TCAM entries, it is possible to see whether any TCAM-allocation failures have occurred. For example, the output of the show platform ip unicast counts command, displayed in Example 7-24, shows if any TCAM-allocation failures were experienced for IP Version 4 prefixes. In general, TCAM-allocation failures are rare because switches have more than enough TCAM capacity for the roles that they are designed and positioned for. However, all networks are different, so be aware of the fact that TCAM-allocation failures can be a possible cause of performance problems.

```
CSW1# show platform ip unicast counts
# of HL3U fibs 141
# of HL3U adjs 9
# of HL3U mpaths 2
# of HL3U covering-fibs 0
# of HL3U fibs with adj failures 0
Fibs of Prefix length 0, with TCAM fails: 0
Fibs of Prefix length 1, with TCAM fails: 0
Fibs of Prefix length 2, with TCAM fails: 0
Fibs of Prefix length 3, with TCAM fails: 0
Fibs of Prefix length 4, with TCAM fails: 0
Fibs of Prefix length 5, with TCAM fails: 0
Fibs of Prefix length 6, with TCAM
```

Another way to spot potential TCAM-allocation failures is by observing traffic being punted to the CPU for forwarding. The command **show controllers cpu-interface** displays packet counts for packets that are forwarded to the CPU. If the retrieved packet counter in the **sw forwarding** row is rapidly increasing when you execute this command multiple times in a row, traffic is being switched in software by the CPU rather than in hardware by the TCAM. An increased CPU load usually accompanies this behavior.

```
CSW1#sh controllers cpu-interface
ASIC    Rxbiterr   Rxunder    Fwdctfix   Txbuflos   Rxbufloc   Rxbufdrain
------------------------------------------------------------------------
ASIC0     0          0          0          0          0          0

cpu-queue-frames  retrieved  dropped    invalid    hol-block  stray
----------------  ---------  -------    -------    ---------  -----
rpc               1          0          0          0          0
stp               853663     0          0          0          0
ipc               0          0          0          0          0
routing protocol  1580429    0          0          0          0
L2 protocol       22004      0          0          0          0
remote console    0          0          0          0          0
sw forwarding     1380174    0          0          0          0
```

One remedy to the TCAM utilization and exhaustion problem is reducing the amount of information that the control plane feeds into the TCAM. For example, you can make use of techniques such as route summarization, route filtering, and access list (prefix list) optimization. Generally, TCAM is not upgradeable, so either the information that needs to be programmed into the TCAM needs to be reduced or you will have to upgrade to a higher-level switch, which can handle more TCAM entries.

**Port utilization:**

You now use the show controller utilization command to check the bandwidth utilization on the ports connecting to the server (port G0/5) and the client (port G0/2). This can help you verify that you indeed have a performance issue. The large discrepancy in the receive and transmit utilization on the user port is due to the fact that the traffic is mostly file downloads. The user is receiving much more than he is sending.

```
Switch# sh controller g0/5 utilization

Receive Bandwidth Percentage Utilization : 0
Transmit Bandwidth Percentage Utilization : 0

Switch# sh controller g0/2 utilization

Receive Bandwidth Percentage Utilization : 2
Transmit Bandwidth Percentage Utilization : 76
```

The first command is show interface accounting, which shows you what kind of traffic is going through the interface. The output shows some STP packets, Cisco Discovery Protocol (CDP) packets, and other packets. There is not a lot of activity, so you do not expect a loop or spanning-tree issue. The traffic bottleneck must come from data itself.

```
Switch# clear counters g0/2
Clear "show interface" counters on this interface [confirm]
Switch#
Switch# sh int g0/2 accounting
GigabitEthernet0/2 to new PC
                Protocol      PktsIn CharsIn PktsOut CharsOut
                Other         0      0       6       360
                Spanning Tree 0      0       32      1920
                CDP           0      0       1       397
```

To find out why the user application is slow, use the **show interface g0/2 stats** command

## Switch Performance Troubleshooting Example: Excessive Broadcasts

The second switch performance troubleshooting example concerns a user with complaints about connectivity issues. The user reports that sometimes he cannot connect to the network at all, and his PC will not even get an IP address. Other times, he is able to connect, but the connection is of poor quality (experiencing slow downloads and connection timeouts).

```
Switch# show processes cpu
CPU utilization for five seconds: 98%/18%; one minute, 5 minute 92%
```

The output displays that CPU utilization is 98 percent over 5 seconds, 94 percent over 1 minute, and 92 percent over 5 minutes! That is very high: the switch is definitely overloaded, and you need to find out why.

Next, use the show processes cpu sorted command, which classifies the processes by task and CPU consumption, to discover the processes that use up most of the CPU cycles. The results shown reveal that ARP is consuming half of the resources on this switch. That is definitely not normal. Having a certain amount of ARP processing is normal, but not to the point where it is consuming more than 40 percent of the resources. Using the command show interfaces accounting, you discover that vlan10 is the where the excessive ARP packets are located.

```
CSW1#show processes cpu sorted
CPU utilization for five seconds: 23%/18%; one minute: 24%; five minutes: 17%
! 23%, 24%, and 17% indicate total CPU spent on processes and interrupts
  (packet switching). 18% indicates CPU spent on interrupts (packet switching)
PID  Runtime(ms)  Invoked      uSecs   5Sec   1Min   5Min  TTY  Process
170       384912  1632941        235  0.47%  0.35%  0.23%    0  IP Input
 63         8462  5449551          1  0.31%  0.52%  0.33%    0  HLFM address
lea
274       101766  1410665         72  0.15%  0.07%  0.04%    0  HSRP IPv4
  4       156599    21649       7233  0.00%  0.07%  0.05%    0  Check heaps

Switch# show interface g0/2 accounting
GigabitEthernet0/2 to new PC
                Protocol     Pkts In   Chars In   Pkts Out   Chars Out
                   Other           0          0          6         360
          Spanning Tree            0          0         32        1920
                    CDP            0          0          1         397
```

To find out which of these ports is the source of the excessive ARP packets, use the show interfaces interface controller include broadcastscommand. This command, with the include broadcasts parameter, displays the broadcast section of the output only. The results, shown in Example 7-37, point to g0/11 and g0/13 ports, to which the wireless access points (WAPs) are connected. You now know that these are the broadcasts from the wireless clients, and because the WAPs act like hubs, they forward all their client broadcasts to the switch.

```
Switch# show interfaces g0/2 controller | inc broadcast
       Received 236 broadcasts (28 multicasts)
Switch# show interfaces g0/9 controller | inc broadcast
       Received 0 broadcasts (0 multicasts)
Switch# show interfaces g0/11 controller | inc broadcast
       Received 2829685 broadcasts (2638882 multicasts)
Switch# show interfaces g0/13 controller | inc broadcast
       Received 41685559 broadcasts (145888 multicasts)
Switch# show interfaces g0/22 controller | inc broadcast
       Received 0 broadcasts (0 multicasts)
```

To reduce the impact of the wireless broadcast on the wired network, you can limit the amount of broadcasts the switch accepts from those ports. As demonstrated in Example 7-38, you use the storm control command on g0/11 and g0/13 interfaces to limit broadcasts (because ARP requests are broadcasts) to three packets per second. Next, you observe the positive results on the output of the show processes cpu sorted command. You confirm with the users that they are no longer experiencing any problems and document your work.

```
Switch(config)# int g0/11
Switch(config-if)# storm-control broadcast level pps 3
Switch(config-if)# int g0/13
Switch(config-if)# storm-control broadcast level pps 3
```

## Switch Performance Troubleshooting Example: Excessive Security

The third and final switch performance troubleshooting example is about a case where users connecting to a specific switch have connectivity issues and say that while working with their PCs a window sometimes pops up indicating that their network cable is unplugged. At other times, the PC reports that the cable is plugged in, but the connection is very bad. Many of the user workstations cannot even obtain an IP address from the DHCP server. Those who do receive IP addresses find the network unusable. Almost all users connected to this switch experience the same problem.

You must remember that the user reported that the connection is intermittent, so although it might be connected now, it might have been disconnected a moment ago. Therefore, you reset the counters on the interface.

```
Switch# clear counters
```

Next, knowing that security policies can be implemented at Layer 2 using VLAN filters, you check if a VLAN filter is applied to VLAN 10. The `show vlan filter vlan 10` command output reveals that a filter called VLAN10_OUT is applied to VLAN 10. Naturally, you display this filter using the show vlan access-map VLAN10_OUT command, so you can analyze it.

```
Switch# sh vlan filter vlan 10
vlan 10 has filter VLAN10_OUT

Switch# sh vlan access-map VLAN10_OUT
Vlan access-map "VLAN10_OUT" 10
Match clauses:
ip address: VLAN10_OUT
Action:
Forward

Vlan access-map "VLAN10_OUT" 20
Match clauses:
ip address: VLAN11_OUT
Action:
Forward

Vlan access-map "VLAN10_OUT" 30
Match clauses:
ip address: VLAN12_OUT VLAN13_OUT VLAN14_OUT VLAN15_OUT
Action:
forward
```

You can see that all of the access maps match on IP address, so this would not have an effect on Layer 1 or2. To be sure, display one of these access lists . You get overwhelmed to see that the access list has more than 400 entries! The situation is made worse by the fact that several access lists are referenced for the packets going into or out of this VLAN.

```
Switch# sh access-list VLAN10_OUT
Extended IP access list VLAN10_OUT
2 permit tcp 10.1.20.0 0.0.0.255 host 10.10.50.124 eq domain
10 permit tcp 10.1.1.0 0.0.0.255 host 10.10.150.24 eq www
11 permit tcp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq www
20 permit tcp 10.1.1.0 0.0.0.255 host 10.10.150.24 eq 22
21 permit tcp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq 22
30 permit tcp 10.1.1.0 0.0.0.255 host 10.10.150.24 eq telnet
31 permit tcp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq telnet
40 permit tcp 10.1.1.0 0.0.0.255 host 10.10.150.24 eq 443
41 permit tcp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq 443
50 permit udp 10.1.1.0 0.0.0.255 host 10.10.150.24 eq snmp
51 permit udp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq snmp
60 permit udp 10.1.1.0 0.0.0.255 host 10.10.150.24 eq snmptrap
61 permit udp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq snmptrap
70 permit tcp 10.1.1.0 0.0.0.255 host 10.10.150.24 eq ftp
71 permit tcp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq ftp
80 permit tcp 10.1.1.0 0.0.0.255 host 10.10.150.24 eq ftp-data
81 permit tcp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq ftp-data
90 permit tcp 10.1.1.0 0.0.0.255 host 10.10.150.24 eq domain
91 permit tcp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq domain
! Output omitted for brevity
```

Next, check to see whether an IP access list is applied to the VLAN 10 interface, using the command show ip interface vlan 10. The output, reveals both an outgoing and an inbound access list VLAN10 applied to the VLAN 10 interface.

```
Switch# sh ip int vlan 10
Vlan10 is up, line protocol is up
Internet address is 10.1.1.1/24
Broadcast address is 255.255.255.255
Address determined by nonvolatile memory
MTU is 1500 bytes
Helper address is not set
Directed broadcast forwarding is disabled
Outgoing access list is VLAN10
Inbound access list is VLAN10
```

When you display access list VLAN 10, you observe a huge output similar to the output when you displayed access list VLAN10_OUT . You cannot help but to wonder if this access list is affecting switch performance so badly that users cannot connect.

```
Switch# sh access-li VLAN10 >> to show access-lists applied to interface vlan10
Extended IP access list VLAN10_OUT
10 permit tcp 10.1.1.0 0.0.0.255 host 10.10.50.24 eq www
11 permit tcp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq www
20 permit tcp 10.1.1.0 0.0.0.255 host 10.10.50.24 eq 22
21 permit tcp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq 22
30 permit tcp 10.1.1.0 0.0.0.255 host 10.10.50.24 eq telnet
31 permit tcp 10.1.1.0 0.0.0.255 host 10.10.151.24 eq telnet
40 permit tcp 10.1.1.0 0.0.0.255 host 10.10.50.24 eq 443
! Output omitted for brevity
```

You know that access lists are managed by the TCAM. So, access list entries should not be managed by the CPU. However, if the TCAM is full, packets will be sent to the CPU for processing (punt). You can verify this using the **show platform tcam utilization** command.

```
Switch# show platform tcam utilization

CAM utilization for ASIC# 0                 Max                  Used
                                        Masks/Values         Masks/Values

Unicast mac addresses:                   6364/6364            29/29
IPv4 IGMP groups + multicast routes:     1120/1120             1/1
IPv4 unicast directly-connected routes:  6144/6144             5/5
IPv4 unicast indirectly-connected routes: 2048/2048           39/39
IPv4 policy based routing aces:           452/452             12/12
IPv4 qos aces:                            512/512              8/8
IPv4 security aces:                       964/964            790/790
```

The IPv4 security access line is eye-catching. There are 964 slots, and 790 slots are in use. Checking the CPU utilization next, you find that it is very high. This indicates that the TCAM is sending packets to the CPU for processing, overloading the CPU as a result.

You have found the problem and its source. The solution, noting that this is an extreme example, is that you need to rewrite and simplify the access lists. Also, you need to verify whether the same VLAN access lists at both the VLAN level and the interface level are necessary. If the access lists cannot be simplified, it might be time to invest in a dedicated platform for security filtering of this network.

**Troubleshooting High CPU Usage Issues on Routers:**

The CPU on a router performs two major tasks: forwarding packets and executing management and control plane processes. The CPU can become too busy when the CPU either has many packets to forward or when a system process consumes a large amount of the CPU time. For example, if the CPU is receiving many SNMP packets because of intensive network monitoring, it can become so busy processing all those packets that the other system processes cannot get access to CPU resources.

The following are some of the most common router processes that could cause high CPU utilization:

**ARP Input:** High CPU utilization by the ARP Input process occurs if the router has to originate an excessive number of ARP requests. Multiple ARP requests for the same IP address are rate-limited to one request every 2 seconds, so excessive numbers of ARP requests can only occur if the router needs to originate ARP requests for many different IP addresses. This can happen if an IP route has been configured pointing to a broadcast interface. This causes the router to generate an ARP request for each IP address that is not reachable through a more specific route. An excessive amount of ARP requests can also be caused by malicious network traffic. An indication of such traffic is the presence of a high number of incomplete ARP entries in the ARP table, similar to the one shown

```
Router# show arp
Protocol Address      Age (min) Hardware Addr Type Interface
Internet 10.10.10.1     -       0013.1918.caae ARPA FastEthernet0/0
Internet 10.16.243.249  0       Incomplete     ARPA
Internet 10.16.243.250  0       Incomplete     ARPA
Internet 10.16.243.251  0       Incomplete     ARPA
Internet 10.16.243.252  0       Incomplete     ARPA
Internet 10.16.243.253  0       Incomplete     ARPA
Internet 10.16.243.254  0       Incomplete     ARPA
```

•

**TCP Timer:** The TCP Timer process is responsible for TCP sessions running on the router. When the TCP timer process uses a lot of CPU resources, this indicates that there are too many TCP peers (such as Border Gateway Protocol [BGP] peers). The **show tcp statistics** command.

```
Router# show tcp statistics
```

**IP Background:** This process is responsible for encapsulation type changes on an interface, the move of an interface to a new state (up or down), and change of IP address on an interface. The IP Background process modifies the routing table in accordance with the status of the interfaces and notifies all routing protocols of the status change of each IP interface.

**There are three types of packet switching modes supported by Cisco routers "Also L3 switches":**

• Process switching

• Fast switching

• CEF

The newest switching mode is CEF, and it is the default, preferred, and recommended switching mode. It is important to remember that the switching method used affects the router's performance. To successfully troubleshoot problems related to the switching path it is essential to understand which method is used and how it works. The switching method might be altered globally or per interface.

## Process Switching:

Process switching is the oldest mode. When using process switching to forward packets, the router strips the Layer 2 header from an incoming frame, looks up the Layer 3 network address in the routing table for the packet, and then sends the frame with a rewritten Layer 2 header, including a newly computed cyclical redundancy check (CRC) to the outgoing interface. All these operations are performed for each individual frame by the IP Input process that is running on the central CPU. Process switching is configured on an interface by disabling fast switching (and CEF) on that interface. Process switching is the most CPU-intensive method available on Cisco routers. It greatly degrades performance figures such as throughput, jitter, latency, and so on. This method should be used only temporarily as a last resort during troubleshooting.

## Note

To use process switching, fast switching must be disabled using this command:

```
Router(config-if)# no ip route-cache
```

## Fast Switching

After performing a routing table lookup for the first packet destined for particular IP network, the router also initializes the fast-switching cache that is used by the fast-switching process. When subsequent frames to that same destination arrive, a cache lookup is performed and the destination is found in the fast-switching cache. Then the frame is rewritten with the corresponding data link layer header that was stored in the cache, and the frame is sent to the outgoing interface. The interface processor computes the CRC for the frame. Because the cache is destination based, fast switching can provide load sharing on a per-destination basis. Fast switching is less processor intensive than process switching because it uses a cache entry created by the first packet sent to a particular destination. The CPU utilization can go high even when the fast switching method is

used, in a situation that there are a high number of new flows per second. This can happen when a network attack generates too many new flows rapidly.

**Note**

Fast switching is enabled using the following command:

```
Router(config-if)# ip route-cache
```

**Cisco Express Forwarding**

Cisco Express Forwarding (CEF) is the default switching mode on Cisco routers. CEF is less CPU-intensive than fast switching or process switching. CEF is a highly scalable and resilient switching technique. When CEF is enabled, information used for packet forwarding purposes resides in the following two tables:

• **CEF Forwarding Information Base (FIB):** A router that has CEF enabled uses the FIB to make IP destination prefix-based switching decisions. This table is updated after each network change, but only once, and contains all known routes. There is no need to build a route cache by first using process switching for some of the packets. Each change in the IP routing table triggers a similar change in FIB table because it contains all next-hop addresses associated with all network destinations.

• **CEF adjacency table:** The adjacency table contains Layer 2 frame headers for all next hops used by the FIB. These addresses are used to rewrite frame headers for packets that are forwarded by a router.

there is no real advantage on enabling CEF on a L2 only switch because only management traffic (telnet, ssh, snmp) is processed at L3, and this requires main cpu attention (it has to be process switched).

It is important to note that there are several Cisco IOS features that require CEF to be enabled for their operation because they rely on the data structures that are built and maintained by Cisco operation. Some of those features are as follows:

• Network-Based Application Recognition (NBAR)

• AutoQoS and Modular QoS CLI (MQC)

• Frame Relay traffic shaping

• Multiprotocol Label Switching (MPLS)

• Class-based weighted random early detection

**Note**

CEF can be enabled and disabled globally using the command:

```
Router(config)# [no] ip cef
```
You can also enable or disable CEF on each interface individually using the command:

```
Router(config-if)# [no] ip route-cache cef
```
Generally, if CEF is disabled globally, it cannot be enabled on an interface, but if it is enabled globally, it can be disabled on a single interface.

**Troubleshooting Process and Fast Switching**

after disabling the default CEF packet-switching mode using the no ip cefcommand. In the output, you can see that fast switching is enabled for all packets (except for packets that are sent back to the same interface that they came in on), but CEF switching is disabled.

If you turn fast switching off, too, using the command no ip route-cache, and repeat the show ip interface command. As you can see, however, multicast fast switching is still enabled. This is because IP multicast routing is configured entirely separate from IP unicast routing and there are separate configuration statements related to unicast and multicast operations. The no ip route-cache command only applies to unicast packets. To disable fast switching for multicast packets, the `no ip mroute-cache` command is used.

```
Router# show ip interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
<...output omitted...>
IP fast switching is enabled
IP fast switching on the same interface is disabled
IP Flow switching is disabled
IP CEF switching is disabled
IP Fast switching turbo vector
IP multicast fast switching is enabled
IP multicast distributed fast switching is disabled
IP route-cache flags are Fast
! Output omitted for brevity
```
The `show ip cache` command displays the content of the fast-switching cache,

**Troubleshooting CEF**

The items that you should check and verify when troubleshooting CEF are as follows:

• Is CEF enabled globally and per interface?

• Is there a FIB entry for a given network destination?

• Is there a next hop associated with this entry?

• Is there an adjacency entry for this next hop?

the `show ip cef` command. This command displays the content of the FIB table All directly connected networks in the output are marked as **attached** in the Next Hop field. Network prefixes that are local to the router are marked as **receive**.

```
Router# show ip cef
Prefix              Next Hop        Interface
0.0.0.0/0           10.14.14.19     GigabitEthernet0/0
0.0.0.0/32          receive
10.14.14.0/24       attached        GigabitEthernet0/0
10.14.14.0/32       receive
10.14.14.252/32     receive
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32  receive
```

You can also see what other destinations are associated with this interface/next-hop pair, using the show ip cef adjacency command for this interface and next-hop value

```
Router# show ip cef adjacency GigabitEthernet0/0 10.14.14.19 detail
```

The Layer2 MAC address for this next-hop IP address can also be checked in the ARP cache using the show ip arp command

```
Router# show ip arp 10.14.14.19
```

You can gather information about the packets that are not switched with CEF by using

```
show cef not-cef-switched
```

**Troubleshooting Router Memory Issues:**

Memory-allocation failure is the most common router memory issue. Memory-allocation failures happen when the router has used all available memory (temporarily or permanently), or the memory has been fragmented into such small pieces that the router cannot find a usable available block. This can happen to the processor memory, which is used by Cisco IOS Software, or to the packet memory, which is used to buffer incoming and outgoing packets. Symptoms of memory allocation failures include the following:

• Messages such as `%SYS-2-MALLOCFAIL: Memory allocation of 1028 bytes failed from 0x6015EC84, Pool Processor, alignment 0 display` in the router logs.

• show commands generate no output.

• Receiving `Low on memory` messages.

• Receiving the `message Unable to create EXEC – no memory or too many processes` on the console.

Some of the main reasons for memory problems are as follows:

**- Memory size does not support the Cisco IOS Software image**

- **Memory-leak bug:** A memory leak occurs when a process requests or allocates memory and then forgets to free (de-allocate) the memory when it is finished with that task. As a result, the memory block stays reserved until the router is reloaded. The **show memory allocating-process totals** command will help you to identify how much memory is used and is free, and the per-process memory utilization of the router. Memory leaks are caused by bugs in the Cisco IOS code, and the only solution is to upgrade Cisco IOS Software on the device to a version that fixes the issue.

```
Router# show memory allocating-process totals
                Head        Total (b)    Used(b)     Free(b)     Lowest(b)  Largest(b)
Processor    62A2B2D0   183323952   26507580   156816372  155132764  154650100
       I/O    ED900000    40894464    4957092    35937372   35887920    3590524
Allocator PC Summary for: Processor
     PC           Total   Count  Name
0x6136A5A8     5234828       1  Init
0x608E2208     3576048     812  TTY data
0x6053ECEC     1557568     184  Process Stack
0x61356928     1365448      99  Init
! Output omitted for brevity
```

**- Security-related problems:** MALLOCFAIL errors can also be caused by a security issue, such as a worm or virus operating in your network.

**- Memory-allocation failure at process = interrupt level:** The error message identifies the cause If the process is listed as <interrupt level>, as shown in the message that follows, the memoryallocation failure is being caused by a software problem:

```
%SYS-2-MALLOCFAIL: Memory allocation of 68 bytes failed from0x604CEF48, pool
Processor, alignment 0-Process= <interrupt level>, ipl= 3
```

You can use the Bug Toolkit to search for a matching software bug ID (unique bug identification) for this issue. After you have identified the software bug, upgrade to a Cisco IOS Software version that contains the fix to resolve the problem.

**- Buffer Leak:**

```
Router# show interfaces
<...output omitted...>
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:58, output never, output hang never
Last clearing of "show interface" counters never
input queue 76/75, 1250 drops
Output queue 0/40, 0 drops;
```

The show buffers command displays statistics for the buffer pools on the router. The output the example reveals a buffer leak in the middle buffers pool. There are a total of 17602 middle buffers in the router, and only 11 are in the free list. This implies that some process takes all the buffers, but does not return them.

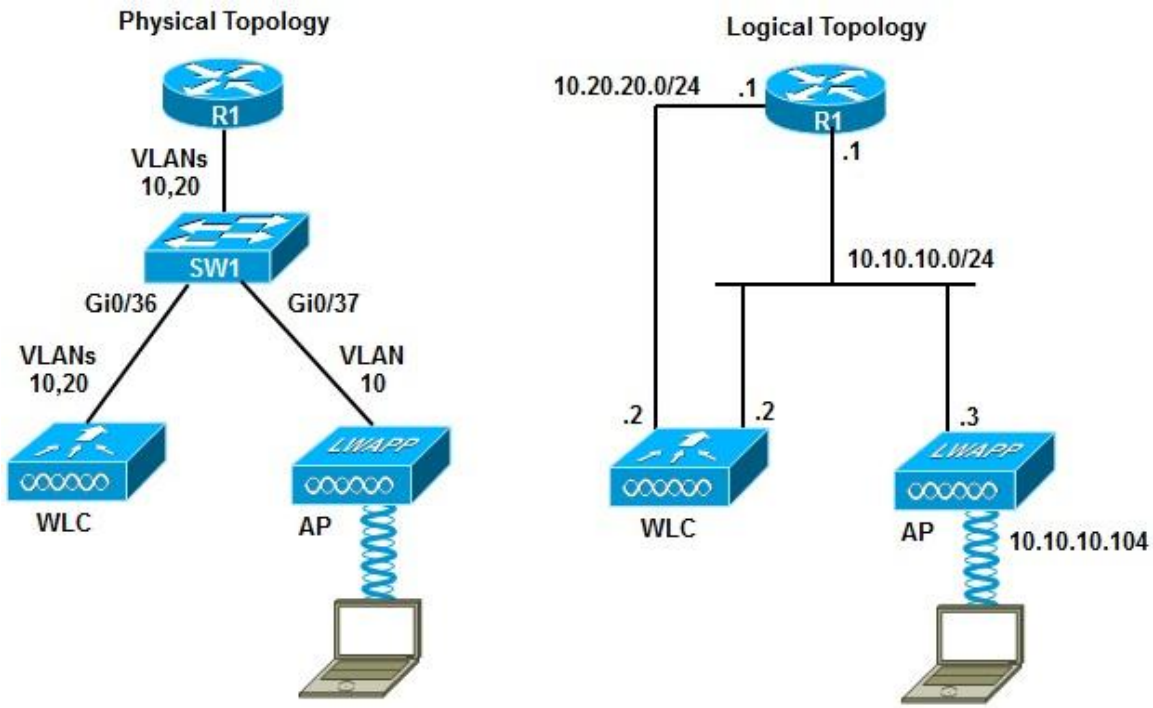
```
Router# show buffers
<...output omitted...>
Middle buffers, 600 bytes (total 17602, permanent 170):
11 in free list (10 min, 400 max allowed)
498598 hits, 148 misses, 671 trims, 657 created
0 failures (0 no memory)
```

If you are using chassis-based routers, which distribute routing information to the line cards, you should not only check the memory availability for the route processor, but also the memory availability on the line cards. The **show diag** command displays the different types of cards present in your router and their respective amounts of memory. This command is useful to identify a lack of memory on the line cards when the router runs BGP

```
Router# show diag | I (DRAM|SLOT)
SLOT 0 (RP/LC 0 ): 1 Port SONET based SRP OC-12c/STM-4 Single Mode
  DRAM size: 268435456 bytes
  FrFab SDRAM size: 134217728 bytes, SDRAM pagesize: 8192 bytes
  ToFab SDRAM size: 134217728 bytes, SDRAM pagesize: 8192 bytes
SLOT 2 (RP/LC 2 ): 12 Port Packet over E3
  DRAM size: 67108864 bytes
  FrFab SDRAM size: 67108864 bytes
  ToFab SDRAM size: 67108864 bytes
SLOT 3 (RP/LC 3 ): 1 Port Gigabit Ethernet
  DRAM size: 134217728 bytes
  FrFab SDRAM size: 134217728 bytes, SDRAM pagesize: 8192 bytes
  ToFab SDRAM size: 134217728 bytes, SDRAM pagesize: 8192 bytes
SLOT 5 (RP/LC 5 ): Route Processor
  DRAM size: 268435456 bytes
```

# Chapter 8. Troubleshooting Converged Networks

## WLAN Connectivity Troubleshooting Example: Misconfigured Trunk




Wireless services have suddenly stopped; clients are not able to associate to the AP. Even from the wired PCs that are used for troubleshooting, it is not possible to connect to the AP or the WLC, using either Secure Shell (SSH) or HTTP-Secure (HTTPS).

```
SW1# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID        Local Intrfce     Holdtme   Capability    Platform     PortID
ap               Gig 0/37          128       T I           AIR-LAP125Gig    0
521-8            Gig 0/39          135                     AIR-LAP521Fas    0
521-7            Gig 0/34          122                     AIR-LAP521Fas    0
Cisco_9a:8c:e0   Gig 0/36          175        H            AIR-WLC210Unit - 0
Slot - 0Port - 1
```

WLC connects to interface Gi 0/36 and the AP connects to interface Gi0/37.

```
SW# show interface status

Port     Name      Status       vlan     Duplex     Speed          Type
Gi0/1              notconnect   1        auto       auto      10/100/1000BaseTX
Gi0/2              notconnect   1        auto       auto      10/100/1000BaseTX
! output omitted for brevity
Gi0/36             connected    trunk    a-full     a-100     10/100/1000BaseTX
Gi0/37             connected    10        a-full     a-1000    10/100/1000BaseTX
```

The Gi 0/37 interface connected to the AP is associated to VLAN 10, and the Gi 0/36 interface connected to the WLC is configured as trunk.

You need to find out which VLANs are used for AP to WLC communication, which VLAN is used for client traffic, and whether the access point is operational and registering to the WLC using LWAPP or Control and Provisioning of Wireless Access Points (CAPWAP). The wireless administrator informs us that the AP has a static IP address and that the WLC and the AP should be on the same VLAN, but the WLC is not seeing registration requests from the AP.

therefore pursue the AP to WLC registration process; if the AP cannot register with the WLC, it will not be able to service client requests. Using the **debug ip packet** command on the switch, you can observe AP's registration process. Because **debug** might produce too much output, you would filter it to see only the traffic related to LWAPP using access list 100. LWAPP uses UDP port 12223 for control messages. The AP's request originating from interface Gi 0/37, which is associated to VLAN 10, is present (see **rcvd** at the end of the**debug** output), but this traffic is not forwarded to the trunk on the Gi 0/36 interface. That is probably why there is no response from the WLC.

```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# access-list 100 permit udp any any eq 12223
SW1(config)# end
SW1# debug ip packet 100
IP packet debugging is on for access list 100
SW1#
5d13h: %SYS-5-CONFIG_I: Configured from console by console
SW1#
5d13h: IP: s=10.80.1.30 (vlan10), d=255.255.255.255, len 123, rcvd 1
5d13h: IP: s=10.10.10.104 (vlan10), d=255.255.255.255, len 123, rcvd 1
5d13h: IP: s=10.10.10.104 (vlan10), d=255.255.255.255, len 123, rcvd 1
```

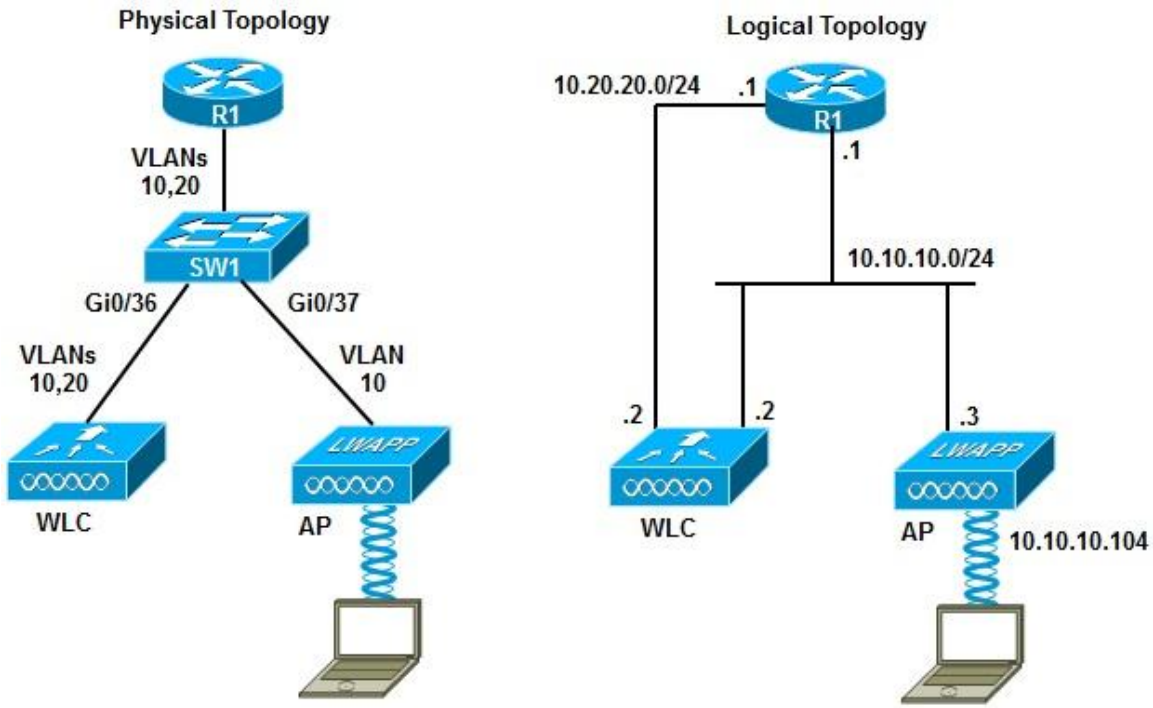
verify that VLAN 10 is allowed on the trunk interface (Gi 0/36)

```
SW1# show interfaces switchport | begin 0/36
Name: Gi0/36
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (Inactive)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan native VLAN: none
Administrative private-vlan Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: 1
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: All
```

You should add the appropriate VLANs to the list of allowed VLANs on the trunk interface. The wireless team tells you that the client VLAN is 10, and that the management VLAN is 20.

```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/36
SW1(config-if)# switchport trunk allowed vlan add 10,20
```

**WLAN Connectivity Troubleshooting Example: Duplex and Trust Issues**




the AP interface pointing to the wired network goes up and down intermittently, and when the port is operational, there is a substantial slowdown on Voice over WLAN.

```
SW1# show logging | include 0/34
00:12:00: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on
GigabitEthernet0/34 (not half duplex), with 521-7 FastEthernet0 (half duplex)
00:13:00: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on
GigabitEthernet0/34 (not half duplex), with 521-7 FastEthernet0 (half duplex)
00:14:00: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on
GigabitEthernet0/34 (not half duplex), with 521-7 FastEthernet0 (half duplex)
00:15:00: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on
GigabitEthernet0/34 (not half duplex), with 521-7 FastEthernet0 (half duplex)
00:16:00: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on
GigabitEthernet0/34 (not half duplex), with 521-7 FastEthernet0 (half duplex)
00:17:00: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on
GigabitEthernet0/34 (not half duplex), with 521-7 FastEthernet0 (half duplex)
00:18:19: %SYS-5-CONFIG_I: Configured from console by console
00:19:00: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on
GigabitEthernet0/34 (not half duplex), with 521-7 FastEthernet0 (half duplex)
00:20:00: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on
GigabitEthernet0/34 (not half duplex), with 521-7 FastEthernet0 (half duplex)
```

```
SW1# show logging
```

```
Syslog logging: enabled (0 messages dropped, 1 messages rate-limited, 0
flushes,
0 overruns, xml disabled, filtering disabled
Console logging: disabled

SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/34
SW1(config-if)# duplex full
SW1(config-if)# speed 100
```

Next, you call the wireless team, and they inform you that the AP comes up and does not go down again, but they are still experiencing performance issues, especially for VoIP traffic coming from the wireless network. When you show processes the CPU is normal.

Next, you consider possible QoS configuration errors. This sounds likely as a possibility; perhaps wireless voice traffic is not being properly prioritized when entering the network. In other words, the voice traffic may not be tagged with proper QoS priorities.

At the switch, use the show mls qos int gi0/34 command to display the trust boundary settings. A trust boundary is the point within the network where QoS markings such as DSCP are first accepted. By default, switch ports will reset DSCP values unless you explicitly tell the port to trust those values. The output of show mls qos indicates that the switch does not trust anything coming from the AP. This could be a real issue; voice traffic is being prioritized on the wireless network, but it is losing its priority when crossing over to the wired network. In the case of a LWAPP deployment, if the AP is tagging packets with values, it is the differentiated services code point (DSCP) field that gets used.

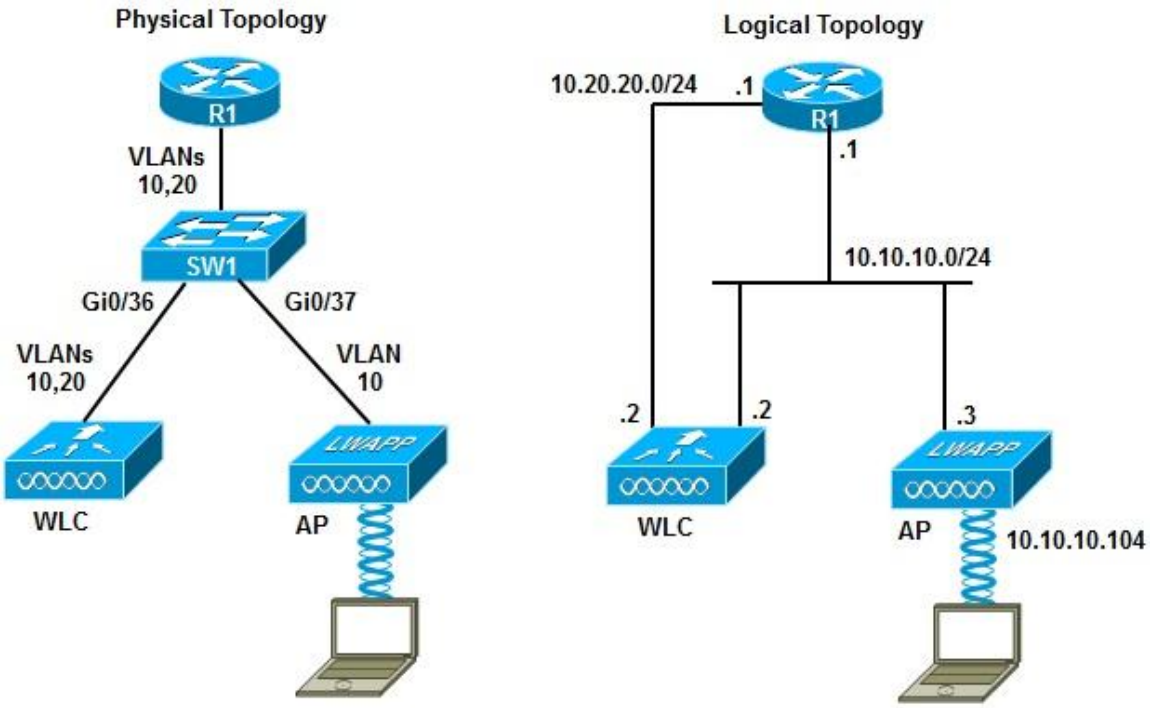
```
SW1# show mls qos int g0/34
GigabitEthernet0/34
trust state: not trusted
trust mode: not trusted
trust enabled flag: ena
COS override: dis
default COS: 0
DSCP Mutation Map: Default DSCP Mutation Map
Trust device: None
qos mode: port-based
```

You must set the switch port to trust DSCP values (following best practices and guidelines) using the mls qos trust dscp command, which is entered at the interface configuration mode. After entering the command, you inspect the configuration with the show mls qos command. The output indicates that the switch is now trusting DSCP values.

```
SW1(config)# int g0/34
SW1(config-if)# mls qos trust dscp
SW1(config-if)# end
SW1# end
SW1#
SW1# show mls qos int g0/34
GigabitEthernet0/34
trust state: trust dscp
trust mode: trust dscp
trust enabled flag: ena
COS override: dis
default COS: 0
```

```
DSCP Mutation Map: Default DSCP Mutation Map
Trust device: None
qos mode: port-based
```

**WLAN Connectivity Troubleshooting Example: LWAPP Denied by New Security Implementations:**




The wireless team tells you that wireless operations have stopped and that none of the APs are able to register to the WLC. This problem has been expected, because a security auditor recently performed a security assessment and recommended a few improvements to the network policy. In taking all the necessary precautions, all configurations have been reverted to their pre-audit state, except for the LAN switch

After investigating the recent change in security policy, you find that Cisco IOS firewall services were installed in some switches that are critical to the network. The auditor recommended hardening network devices at locations subject to higher levels of risk. Having studied the security implications and considerations on a wireless network, you know that the wireless-to-wired edge is a likely candidate to have a firewall deployed. This line of thinking allows you to focus on the Cisco IOS firewall, without discarding the possibility of other issues. Therefore, instead of focusing on a bottom-up or top-down approach, you start at the firewall level and analyze the implications of it in the wireless infrastructure. The reported symptom, wireless APs not being able to register to the WLC, provides another hint as to what to look for: LWAPP traffic may be denied by the firewall. This is a valid hypothesis with a good likelihood of being accurate, and you need

to verify it. While gathering information about the Cisco IOS firewall, you must remember that Cisco IOS Software allows the firewall to be configured using one of two methods:

• The classical Cisco IOS firewall, which uses ACLs exclusively on interfaces

• The zone-based firewall, more widely used and more flexible for a comprehensive deployment of

firewall rules You check the zone-based policy first, but after entering the show zone-pair security command you receive an error message, effectively informing you that no zone-based policies are configured on this router. The show zone-pair security command is normally used to display the policy attached to zone-pairs.

```
R1# show ip interface g0/34
GigabitEthernet0/34 is up, line protocol is up
Inbound access list is FIREWALL
```

```
R1# show access-list
Extended IP access list 100
  10 permit udp 10.10.10.0 0.0.0.255 any eq 12223
  20 permit udp any any eq 12223
Extended IP access list FIREWALL
  10 permit icmp any any echo-reply
  20 permit tcp any any eq www
  30 permit tcp any any eq ftp
  40 permit tcp any any eq ftp-data
  50 permit tcp any any eq telnet
  60 permit tcp any any eq smtp
  70 permit tcp any any eq pop3
  80 permit eigrp any any
  90 permit udp any any eq rip
```

You need to permit UDP 12222 for user data traffic, and UDP 12223 for AP-to-WLC control messages.

```
R1(config)# ip access-list extended FIREWALL
R1(config-ext-nacl)# remark ---allowing LWAPP control and data ports---
R1(config-ext-nacl)# permit udp any any range 12222 12223
```

## WLAN Connectivity Troubleshooting Example: DHCP Issues

1- When you use an L3 switch SVI as the default gateway for certain vlan, do not forget to configure the ip helper address for the dhcp on the SVI to enable the switch to send the DHCP requests to the DHCP server.

2- Lightweight Access Points cannot register into the WLC. The wireless operations team tells you to check the configuration of option 43 on the DHCP server. On the DHCP server, you display the details of the address pool using the **show ip dhcp pool** command and see nothing there. The **show**

**running | section ip dhcp pool** displays no option 43 either. Option 43 is used to notify the DHCP client the AP-management IP address of the WLC.

For that, you will use the command option 43, followed by the right IP address in hexadecimal format. The hex string is assembled by concatenating Type, Length, and Value. Type is always f1 (hex), Length is the number of controller management IP addresses times 4 in hex, and Value is the IP address of the controller listed sequentially in hex. If there is only one WLC management address, the Length is 04 (hex), and in this case the WLC management IP address is 10.10.10.10, which is 0a0a0a0a (hex).

```
R1(config)# ip dhcp pool vlan10
R1(dhcp-config)# option 43 hex f1040a0a0a0a
```

Convert the ip to decimal then convert it to hex.

# Troubleshooting Unified Communications Issues in a Converged Network

Useful Converged Network Troubleshooting Commands

### Switching

```
show interfaces trunk
show interfaces switchport
show vlan
show errdisable recovery
```

### Auto-QoS

```
show auto qos show auto discovery qos
```

### IP services

```
show ip dhcp pool
show ip dhcp server
show ntp status
```

### IP communications

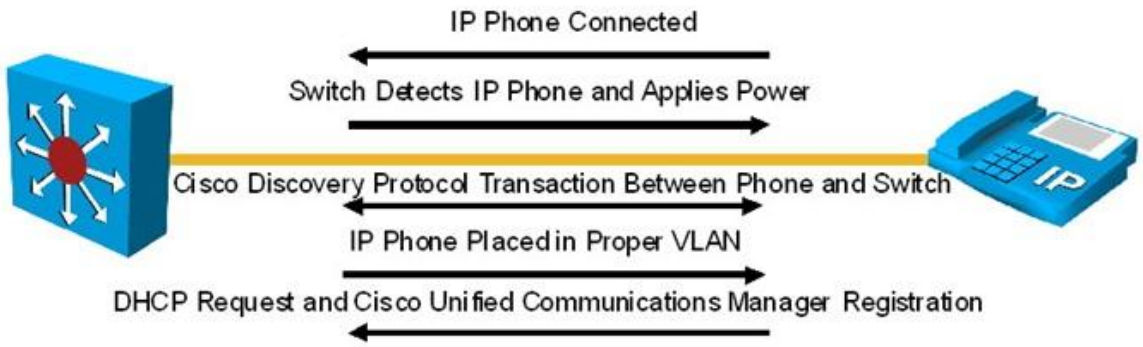
```
debug ephone
```

### Security

```
 show crypto engine connections active
```

**IP phone boot process**

One of the important processes in the network that the support engineers need to be familiar with is the IP phone boot process. Several devices, services, and protocols need to work in harmony for the successful initialization and startup of the IP phones (see Figure 8-7). Knowing the process is critical to laying out an effective troubleshooting method and making good use of the available tools and commands. The following is a list of IP phone boot process steps:




Step 1. The IP phone powers on.

Step 2. The phone performs a power-on self-test, or POST.

Step 3. The phone boots.

Step 4. The phone uses CDP to learn the voice VLAN.

Step 5. The phone initializes the IP stack.

Step 6. The IP phone sends DHCP broadcasts.

Step 7. The DHCP server selects a free IP address from the pool and sends it, along with the other parameters, including option 150.

Step 8. The IP phone initializes, applying the IP configuration to the IP stack.

Step 9. The IP phone requests a configuration file from the TFTP server defined in Option 150.

Note that prior to IP phone power on in Step 1, the LAN switch to which it connects must detect the phone's Power requirement and apply power (PoE) to the appropriate port accordingly. Furthermore, after the phone copies its configuration file from the TFTP server in Step 9, it registers with the CUCM that the configuration file specifies.

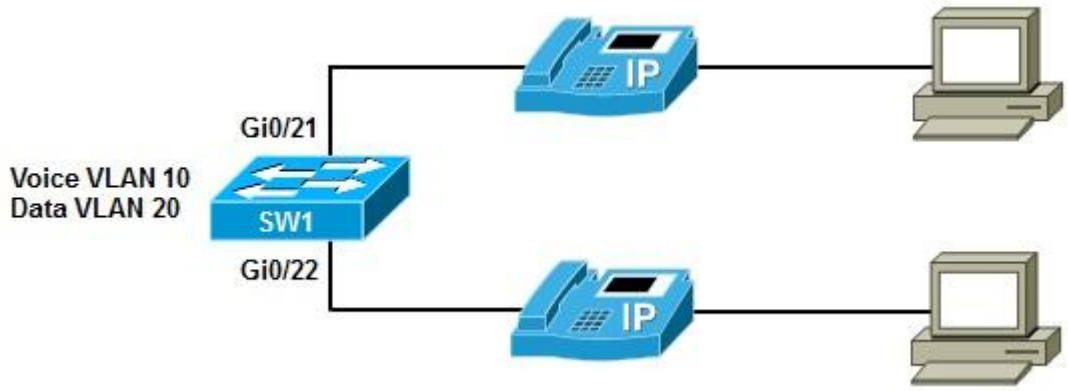
Configuration Example:

data VLAN is 2, and your new voice VLAN is 100

```
Switch# conf t
Switch(config)# int gig 1/0/2
Switch(config-if)# Description *** Port 7A - Phone ***
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 2
Switch(config-if)# switchport voice vlan 100
Switch(config-if)# spanningtree portfast
```

The phone will on boot automatically place itself in vlan 100 for telephony and pass through vlan 2 to the PC port on the phone.

**Troubleshooting Example: Port Security and Voice VLAN Issues**




The problem here is that the IP phones will not boot and initialize. They have no access to the IP network. We are having this problem in multiple areas of the network, but not all of them.

The change logs for the affected wiring closets show a recent change on VLAN Trunking Protocol (VTP) domains and configuration.

```
Switch# show interfaces g0/21 status
Port      Name               Status        Vlan   Duplex  Speed  Type
Gi0/21    to phone number one err-disabled  20     auto    auto   10/100/1000BaseTX
```

The err-disable state can have multiple causes: duplex mismatches, late collisions, EtherChannel problems, spanning-tree issues, and so on.

```
show interface status err-disabled:
```

This command lists the ports in this state along with the reasons for this state.

```
Switch# show interface status err-disabled
Port   Name                 Status        Reason          Err-disabled vlans
Gi0/21 to phone number one  err-disabled  psecure-violation
```

```
Switch# show port-security interface g0/21
Port Security                   : Enabled
Port Status                     : Secure-shutdown
Violation Mode                  : Shutdown
Aging Time                      : 0 mins
Aging Type                      : Absolute
SecureStatic Address Aging      : Disabled
Maximum MAC Addresses           : 1
Total MAC Addresses             : 1
Configured MAC Addresses        : 1
Sticky MAC Addresses            : 0
Last Source Address:vlan        : 0021.7098.30ab:20
Security Violation Count        : 1
```

Some of the phones have PCs connected to them, and both the phone and the PC send packets. This means that two MAC addresses will be reported on the port, which is beyond the maximum allowed.

Note that the port security feature is not necessary on VOIP switch port interfaces according to the company policy:

```
Switch(config)# int g0/21
Switch(config-if)# shutdown
Switch(config-if)# no switchport port-security
Switch(config-if)# no switchport port-security mac-address 000b.8572.1810
Switch(config-if)# no shutdown
```

After the corrections are made, you must reset the interface by entering shutdown before removing the erroneous commands, and entering the no shutdown command after-wards

```
Switch# sh int g0/21 status
Port    Name                Status     Vlan   Duplex  Speed  Type
Gi0/21  to phone number on  connected  20     a-full  a-1000 10/100/1000BaseTX
```

You hear back from the IP telephony support personnel, and they state that their IP phones are still down. So, you must continue troubleshooting. Scrolling back through the running configuration of the interface, you notice that voice VLAN is not configured for the port. At this point, the support team has provided you with the configuration template for switch ports connecting IP phones to the network.

```
Switch(config)# int g0/21
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# mls qos trust cos
Switch(config-if)# mls qos trust device cisco-phone
```

Note:

- The IP phone registers to the router using Skinny Client Control Protocol (SCCP), which is also referred to as “Skinny.” SCCP runs over TCP and uses port 2000, so if you have a firewall using the access-lists make sure that this port is allowed.

- to debug the registration process of an IP-Phone to the CUCM use the following command:

```
SW#debug ephone register
*Sep 1 17:22:37.179: ephone-1[0/1]:SkinnyCompleteRegistration
```

## Troubleshooting Video Issues in a Converged Network

you notice the similarities of this diagram and its components with a voice-enabled network. In fact, the good news is that several components and infrastructure services are shared between video and voice applications. Sometimes the endpoints are the same, or at least integrated. Some of the critical protocols, such as SIP, are going to be the same too. SIP is a signaling protocol that is used to initiate, manage, and terminate voice calls but also video sessions.

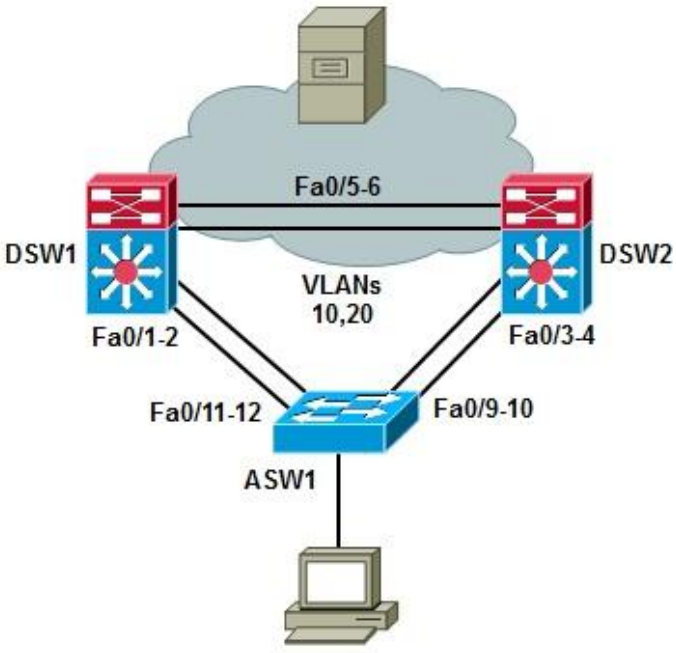
Similar to the voice deployments, the protocols that might need to be permitted are SIP, H.323, SCCP (Skinny), RTP, RTCP, and perhaps some others.

## Video Application QoS Requirements

| Metric | Video Collaboration | Cisco TelePresence | Video Surveillance |
|---|---|---|---|
| Latency | 200 ms | 150 ms | 500 ms |
| Jitter | 10 ms | 10 ms | 10 ms |
| Loss | 0.05% | 0.05% | 0.5% |

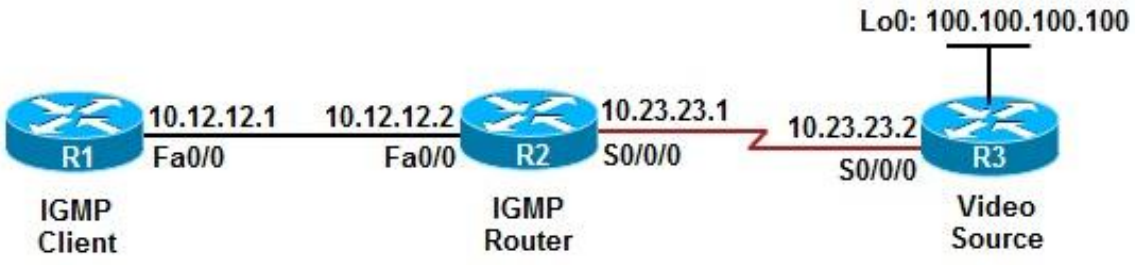
## Troubleshooting Example: Performance Issues Due to STP Topology

Users complain about"poor" performance of their video application. The scenario includes the switched network in the figure, where the video clients reside in two VLANS, 10 and 20, implemented in the access switch. The access switch is serviced by two distribution switches that connect the clients to the campus network, where the video server resides. The distribution switches have recently been upgraded to a new version of Cisco IOS Software. After the change, users started complaining about the poor performance. This is because DSW1 is the root for all vlans and the common STP is used in this case the solution is:




```
DSW1(config)# spanning-tree vlan
10,30,50 root primary
DSW1(config)# spanning-tree vlan 20,40,60 root secondary
DSW2(config)# spanning-tree vlan 10,30,50 root secondary
DSW2(config)# spanning-tree vlan 20,40,60 root primary
```

**Troubleshooting Example: IP Multicast Configuration Error:**




This network is simulating an IGMP network, with R1 acting as an IGMP client, similar to a PC running a video application and joining multicast groups. R2 will act as the first-hop router, listening to IGMP join and leave transactions. And R3 will act as the video server, pushing multicast traffic downstream. The video server is simulated by the loopback interface with the IP address 100.100.100.100 on R3. R2 and R3 are preconfigured to communicate multicast group information through Protocol Independent Multicast (PIM). R1 and R2 are preconfigured to use IGMP to allow R1 to join multicast groups. The problem here is that users in the R1 LAN are not able to watch the video stream; they are able to connect to the server and request the video, but the video stream is not reaching them after that. The application team has verified that the software is installed correctly and the server is configured properly, and they suspect the network is to blame.

Because the video application is the only one that is not working, you can discard IP reachability and routing issues as the problem,

Because you are dealing with a multicast issue, keep in mind that end devices must join a multicast group before they can receive traffic directed to that group. So, you get on R2 to see the multicast groups the hosts in this LAN have joined.

```
R2# show ip igmp group
IGMP Connected Group Membership
Group Address    Interface      Uptime    Expires  Last Reporter Group Accounted
224.0.1.40       Serial0/0/0    00:08:48  Stopped  10.23.23.2
```

R1 is not joining any group. The group that you see on the example output is on the serial 0/0/0 interface, while R1 is on the LAN interface Fa 0/0. The show ip igmp membership command, which shows all members of all groups, does not list the IP address of R1 (10.12.12.1) anywhere.

```
R2# show ip igmp membership
Channel/Group   Reporter      Uptime    Exp   Flags    Interface
*.224.0.1.40    10.23.23.2    00:09:24  stop  2LA      Se0/0/0
```

Because R1 is simulating an IGMP client, you enter the command ip igmp join-group and join the group 224.8.8.8.

```
R1(config)# int fa0/0
R1(config-if)# ip igmp join-group 224.8.8.8
```

The first step you must take to further isolate the problem is to determine whether IGMP is actually running on router R2. So, you use the show ip igmp interface command on R2. There is only one interface enabled for IGMP in this router and that is serial 0/0/0. Because IGMP is not enabled on the R2's Fa0/0 interface, R1 could not join the multicast group, and that is why multicast connectivity is not working.

```
R2# show ip igmp interface
Serial0/0/0 is up, line protocol is up
Internet address is 10.23.23.2/24
IGMP is enabled on interface
Current IGMP host version is 2
Current IGMP router version is 2
IGMP query interval is 60 seconds
IGMP querier timeout is 120 seconds
IGMP max query response time is 10 seconds
Last member query count is 2
Last member query response interval is 1000 ms
Inbound IGMP access group is not set
IGMP activity: 1 joins, 0 leaves
Multicast routing is enabled on interface
Multicast TTL threshold is 0
IGMP querying router is 0.0.0.0 (this system)
Multicast groups joined by this system (number of users):
224.0.1.40(1)
```

You will configure IGMP on router R2's Fa 0/0 interface by enabling PIM on this interface, using the command **ip pim sparse-dense**

```
R1(config)# int fa0/0
R1(config-if)# ip pim sparse-dense-mode
```

```
R2# show ip igmp interface
Serial0/0/0 is up, line protocol is up
Internet address is 10.23.23.2/24
IGMP is enabled on interface
Current IGMP host version is 2
Current IGMP router version is 2
IGMP query interval is 60 seconds
IGMP querier timeout is 120 seconds
IGMP max query response time is 10 seconds
Last member query count is 2
Last member query response interval is 1000 ms
Inbound IGMP access group is not set
IGMP activity: 1 joins, 0 leaves
Multicast routing is enabled on interface
Multicast TTL threshold is 0
IGMP querying router is 0.0.0.0 (this system)
Multicast groups joined by this system (number of users):
```

```
224.0.1.40(1)
FastEthernet0/0 is up
Internet address is 10.12.12.2/24


R2# show ip igmp group
IGMP Connected Group Membership
Group       Interface          Uptime    Expires   Last Reporter
224.8.8.8   FastEthernet0/0/0  00:08:48  00:02:51  10.12.12.1
224.0.1.40  Serial0/0/0        00:19:43  stopped   10.23.23.2
```

Finally, you need to verify that multicast connectivity is working end to end. You go to the video server, R3 in this example, and ping 224.8.8.8. This action simulates multicast traffic originating from the multicast server (R3), which must reach the members of the multicast group 224.8.8.8.

```
R3# ping 224.8.8.8
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.8.8.8, timeout is 2 seconds:
Reply to request 0 from 10.12.12.1, 1 mss
```

# Chapter 9. Maintaining and Troubleshooting Network Security Implementations

Authentication, authorization, and accounting (AAA) is a major component of the network security of the organization. AAA is the basis for providing secure remote access to the network and remote management of network devices. Network devices can use a centralized security server containing all of the security polices that define the list of users and what they are allowed to do. TACACS+ and RADIUS are the commonly used protocols to communicate with a centralized (AAA) security server such as Cisco Secure Access Control Server (ACS). The following are some of the main characteristics of these protocols:

• RADIUS combines authentication and authorization. The access-accept packets sent by the RADIUS server to the client contain authorization information. This makes it difficult to decouple authentication and authorization. On the other hand, TACACS+ uses the AAA architecture, which decouples authentication and authorization.

• RADIUS uses UDP, whereas TACACS+ uses TCP (port 49). RADIUS uses UDP port 1812 (or 1645) for authentication and UDP port 1813 (or 1646) for accounting messages.

• RADIUS encrypts only the password in the access-request packet, from the client to the server. The remainder of the packet is unencrypted. Other information, such as username, authorized services, and accounting, can be captured by a third party. TACACS+ encrypts the entire body of the packet but leaves a standard TACACS+ header. Within this header, a field indicates whether the body is encrypted. For debugging purposes, it is useful to have the body of the packets unencrypted; however, during normal operation, the body of the packet is fully encrypted for more secure communications.

• RADIUS does not allow specification (or enforcement) of which commands can be and which commands cannot be executed on a router on a per user basis. Therefore, RADIUS is not as useful for router management or as flexible for terminal services. TACACS+ provides two methods to control the authorization of router commands on a per-user or per-group basis:

• Assign privilege levels to commands and have the router verify with the TACACS+ server whether the user is authorized at the specified privilege level

• Explicitly specify in the TACACS+ server, on a per-user or per-group basis, the commands that are allowed

• RADIUS has extensive accounting capabilities, whereas TACACS+ has limited accounting capabilities.

• RADIUS is based on an open standard (RFC 2865); TACACS+ was developed by Cisco Systems.

**Securing the Management Plane of CISCO devices:**

From a troubleshooting standpoint, it is important to know the answer to the following questions:

• What security policies have been implemented for management access to the devices?

• From which IP addresses or networks can the network devices be accessed?

• What type of authentication, authorization, and accounting is used on the network?

• If centralized AAA services are deployed, what happens when these servers fail or become unreachable?

• Are there any backdoors or fallback mechanisms to access the devices?

The output of the debug aaa **authentication** command can be very useful for troubleshooting AAA

```
Router# debug tacacs
Router# debug aaa authentication
13:21:20: AAA/AUTHEN: create_user user=’’ ruser=’’ port=’tty6’
rem_addr=’10.0.0.32’ authen_type=1 service=1 priv=1
13:21:20: AAA/AUTHEN/START (0): port= ‘tty6’ list=’’ action=LOGIN service=LOGIN
13:21:20: AAA/AUTHEN/START (0): using “default” list
13:21:20: AAA/AUTHEN/START (70215483): Method=TACACS+
13:21:20: TAC+ (70215483): received authen response status = GETUSER
13:21:20: AAA/AUTHEN (70215483): status = GETUSER
13:21:23: AAA/AUTHEN/CONT (70215483): continue_login
13:21:23: AAA/AUTHEN (70215483): status = GETUSER
13:21:23: AAA/AUTHEN (70215483): Method=TACACS+
13:21:23: TAC+ : send AUTHEN/CONT packet
13:21:23: TAC+ (70215483): received authen response status = GETPASS
13:21:23: AAA/AUTHEN (70215483): status = GETPASS
13:21:27: AAA/AUTHEN/CONT (70215483): continue_login
13:21:27: AAA/AUTHEN (70215483): status = GETPASS
13:21:27: AAA/AUTHEN (70215483): Method=TACACS+
13:21:27: TAC+ : send AUTHEN/CONT packet
13:21:27: TAC+ (70215483): received authen response status = PASS
13:21:27: AAA/AUTHEN (70215483): status = PASS
```

The debug aaa authentication output captured in Example shows the following events:

• A remote user with the IP address 10.0.0.32 attempts to log in to the router.

• The router checks to see whether AAA authentication for the LOGIN service is enabled (and it is).

• Because no other authentication method list is applied, the router applies the default method, and the first method defined by the default method list is TACACS+.

• The authentication process then prompts the user for a username and password, and upon receiving these credentials, they are sent to the TACACS+ server for verification.

• The TACACS+ server checks the credentials against its database and responds with a PASS status as a sign of successful authentication.

**Note**

the lines in the output that are preceded by AAA are generated by the debug aaa authentication command, while the lines preceded by TAC+ are originated by the debug tacacs command.

Authorization determines what resources the user has access to (on the router, for example). In other words, AAA authorization assembles a set of attributes that describe what tasks the user is authorized to perform.

the output of the debug aaa **authorization** command can prove quite useful to troubleshoot AAA authorization problems.

```
Router# debug aaa authorization
2:23:21: AAA/AUTHOR (0): user= 'admin1'
2:23:21: AAA/AUTHOR (0): send AV service=shell
2:23:21: AAA/AUTHOR (0): send AV cmd*
2:23:21: AAA/AUTHOR (342885561): Method=TACACS+
2:23:21: AAA/AUTHOR/TAC+ (342885561): user=admin1
2:23:21: AAA/AUTHOR/TAC+ (342885561): send AV service=shell
2:23:21: AAA/AUTHOR/TAC+ (342885561): send AV cmd*
2:23:21: AAA/AUTHOR (342885561): Post authorization status = FAIL
```

The debug aaa authorization output captured in Example shows the following events:

• The user (admin1) is attempting to do something that requires authorization (in Example , the user attempts to gain an EXEC shell service).

• The cmd parameter specifies a command that the user is trying to execute. If * is listed, it refers to plain EXEC access.

• The method used to authorize this access is TACACS+.

• The router sends the necessary information (user credentials) through TACACS+ to the security server.

• The security server verifies the authorization, determines that the user is not authorized to perform this function, and sends back a FAIL status response.

Accounting provides a method for collecting and sending information about user activities to the security server for billing, auditing, and reporting purposes. This information includes user identities, start and stop times, executed commands (such as PPP), and the number of packets and bytes sent and received. When AAA accounting is activated, the network access server reports user activity to the RADIUS or TACACS+ security server in the form of accounting records.

The debug aaa **accounting** command is very informative; hence, this debug can help you troubleshoot AAA accounting problems.

```
Router# debug aaa accounting
May 10 14:48:33.011: AAA/ACCT/EXEC(00000005): Pick method list 'default'
May 10 14:48:33.011: AAA/ACCT/SETMLIST(00000005): Handle 0, mlist 81CA79CC,
Name default
May 10 14:48:33.011: Getting session id for EXEC (00000005): db=82099258
May 10 14:48:33.011: AAA/ACCT/EXEC(00000005): add, count 2
May 10 14:48:33.011: AAA/ACCT/EVENT(00000005): EXEC UP
```

The debug aaa accounting output captured in Example  shows a scenario where the default method was used, and that a user successfully gained access to the router's EXEC shell.

There are a number of common TACACS+ failures, including the following:

- A TACACS+ server goes down or a device (TACACS client) cannot connect to the server. To be ready for a situation like this, you may want to configure the network device to use the local database for authenticating critical users.

- A device (client) shared key and the server's shared key do not match.

- User credentials (username, password, or both) getting rejected by the server.

```
Router# debug tacacs
Router# debug aaa authentication
! The TACACS+ server is down or the device has no connectivity to the server:
TAC+: TCP/IP open to 171.68.118.101/49 failed-
Connection refused by remote host
AAA/AUTHEN (2546660185): status = ERROR
AAA/AUTHEN/START (2546660185): Method=LOCAL
AAA/AUTHEN (2546660185): status = FAIL
As1 CHAP: Unable to validate response. Username chapuser: Authentication
failure

! The key on the device and TACACS+ server do not match:
TAC+: received bad AUTHEN packet: length = 68, expected 67857
TAC+: Invalid AUTHEN/START packet (check keys)
AAA/AUTHEN (1771887965): status = ERROR

! Bad user name, bad password, or both:
AAA/AUTHEN: free_user (0x170940) user= 'chapuser' ruser= ''
Port= 'Async1' rem_addr= 'async' authen_type=CHAP service=PPP priv=1
TAC+: Closing TCP/IP 0x16EF4C connection to 171.68.118.101/49
AAA/AUTHEN (2082151566): status = FAIL
As1 CHAP: Unable to validate Response. Username papuser: Authentication failure
```

The common RADIUS problems are similar to the common TACACS+ problems. Example cases of RADIUS server's failure or loss of network connectivity, mismatch of the shared key between the RADIUS server and the network device (RADIUS client), user authorization failure, and finally, a case of bad username, password, or both.

```
Router# debug radius
Router# debug aaa authentication
! The RADIUS server is down or the device has no connectivity to the server:
As1 CHAP: I RESPONSE id 12 len 28 from "chapadd"
RADIUS: id 15, requestor hung up.
RADIUS: No response for id 15
RADIUS: No response from server
AAA/AUTHEN (1866705040): status = ERROR
AAA/AUTHEN/START (1866705040): Method=LOCAL
AAA/AUTHEN (1866705040): status = FAIL
As1 CHAP: Unable to validate Response. Username chapadd: Authentication failure
As1 CHAP: O FAILURE id 13 len 26 msg is "Authentication failure"

! The key on the device and RADIUS server do not match:
RADIUS: received from id 21 171.68.118.101:1645, Access-Reject, len 20
RADIUS: Reply for 21 fails decrypt
NT client sends 'DOMAIN\user' and Radius server expects 'user':
RADIUS: received from id 16 171.68.118.101:1645, Access-Reject, len 20
AAA/AUTHEN (2974782384): status = FAIL
As1 CHAP: Unable to validate Response. Username CISCO\chapadd: Authentication
failure
As1 CHAP: O FAILURE id 13 len 26 msg is "Authentication failure"

! Username and password are correct, but authorization failed:
RADIUS: received from id 19 171.68.118.101:1645, Access-Accept, len 20
RADIUS: no appropriate authorization type for user
AAA/AUTHOR (2370106832): Post authorization status = FAIL
AAA/AUTHOR/LCP As1: Denied

! Bad username, bad password, or both:
RADIUS: received from id 17 171.68.118.101:1645, Access-Reject, len 20
AAA/AUTHEN (3898168391): status = FAIL
As1 CHAP: Unable to validate Response. Username ddunlap: Authentication failure
As1 CHAP: O FAILURE id 14 len 26 msg is "Authentication failure"
As1 PPP: Phase is TERMINATING
```

## Securing The Data Plane

The Cisco IOS firewall software provides enhanced security functions for the data plane. There are two types of Cisco IOS firewall:

• Classic Cisco IOS firewall (stateful packet inspection)

• Zone-based policy firewall

### Securing the Data Plane Using IOS Stateful Packet Inspection "SPI":

Cisco IOS stateful packet inspection (SPI) is a component of the Cisco IOS firewall.

Formerly known as context-based access control (CBAC) Cisco SPI allows certain incoming flows by first inspecting and recording flows initiated from the trusted network. It is configured per interface and operates by dynamically modifying access list entries based on traffic flows. IOS SPI can inspect to the application layer The combination of the inspection policy and the ACL-based policy defines the overall firewall policy.

To protect a trusted (internal) network from an untrusted (external) network using a router with two interfaces, the router can be placed between the two networks. There will be four logical points at which the router can inspect traffic:

Inbound on the internal interface

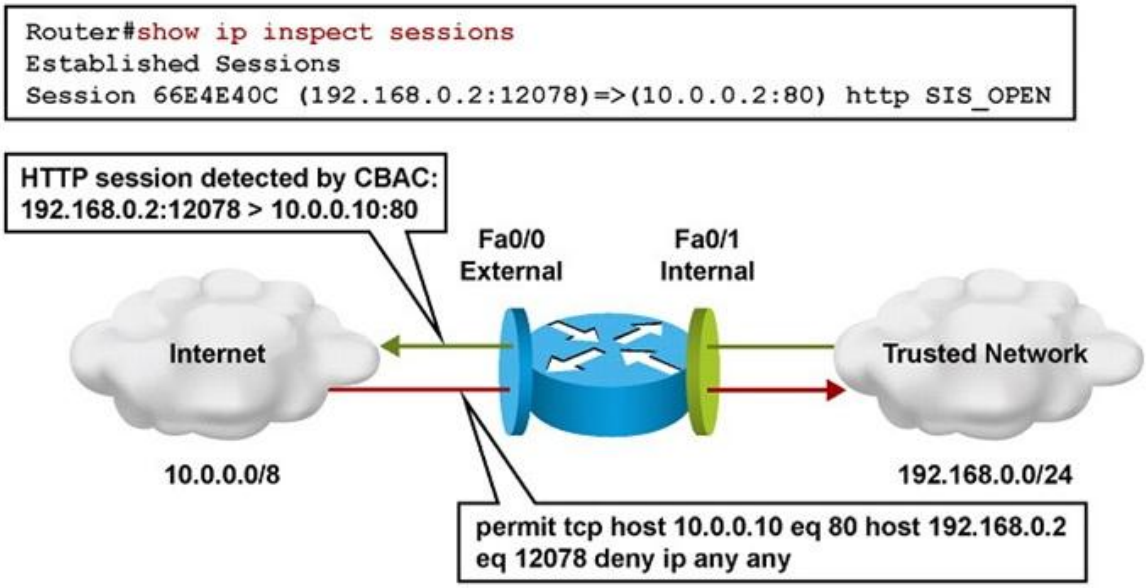
Outbound on the external interface

Inbound on the external interface

Outbound on the internal interface

## IOS SPI Example

In the figure, all traffic from the internal LAN to the Internet is allowed and traffic flowing from the Internet toward the LAN is denied.




First apply a simple access list to deny all IP traffic for the inbound direction of the external interface (Fa 0/0). In this example an ACL Denying All Inbound Traffic Is Created and Applied to Fa 0/0.

```
Router(config)# ip access-list extended DENY_ALL
Router(config-ext-nacl)# deny ip any any
Router(config-ext-nacl)# exit
Router(config)# interface fa0/0
Router(config-if)# ip access-group DENY_ALL in
Router(config-if)# exit
```

To allow responses to return from the Internet for internal client web requests, define an inspection rule to track HTTP sessions.

Create an inspection rule called inshttp to monitor HTTP and apply it outbound on the external interface (Fa0/0).

The router will inspect traffic originating from the trusted network, and dynamically adjust the ACL restricting traffic inbound on the external interface.

```
Router(config)# ip inspect name inshttp http
Router(config)# interface fa0/0
Router(config-if)# ip inspect inshttp out
Router(config-if)# end
```

The output of the **show ip inspect sessions** in the figure shows that trusted host 192.168.0.2 has opened an HTTP connection to external web server 10.0.0.2

Use the **show ip inspect all** command to display the SPI configuration and session information.

```
Router# show ip inspect all
Session audit trail is enabled
Session alert is enabled
<output omitted>
Inspection Rule Configuration
 Inspection name inshttp
    http alert is on audit-trail is on timeout 3600
    https alert is on audit-trail is on timeout 3600
Interface Configuration
 Interface FastEthernet0/0
  Inbound inspection rule is not set
  Outgoing inspection rule is inshttp
    http alert is on audit-trail is on timeout 3600
    https alert is on audit-trail is on timeout 3600
  Inbound access list is DENY_ALL
  Outing access list is not set
<output omitted>
```

An audit trail can be enabled to generate syslog messages for each SPI session creation and deletion using the **ip inspect audit-trail** command. The output of the **debug ip inspect** command provides greater detail.

**An alternative solution to this case is to use a source port number in the access list, for example if you want to allow http traffic from internal network to the internet but in the same time to disallow traffic coming from the internet. Hosts will use random port numbers when communication with the outside for example you tried to browse an http the destination port number will be 80 so if you applied the access list to the outbound in the router local interface to filter the internet response you need to allow the router to allow the port number 80 that the outside will use as the source port when the packet comes back. you may use the following:**

```
Permit any eq 80 any
```

## Zone-Based Policy Firewall Overview

- The zone-based policy firewall (ZPF) is the most current Cisco firewall technology.

- ZPF allows grouping of physical and virtual interfaces.

- Firewall policies are configured on traffic moving between zones.

- ZPF simplifies firewall policy troubleshooting by applying explicit policy on interzone traffic.

- Firewall policy configuration is very flexible.

- Varying policies can be applied to different host groups, based on ACL configuration.

- ZPF supports the following functionalities:

- Stateful inspection

Application inspection: IM, POP, IMAP, SMTP/ESMTP, HTTP URL filtering

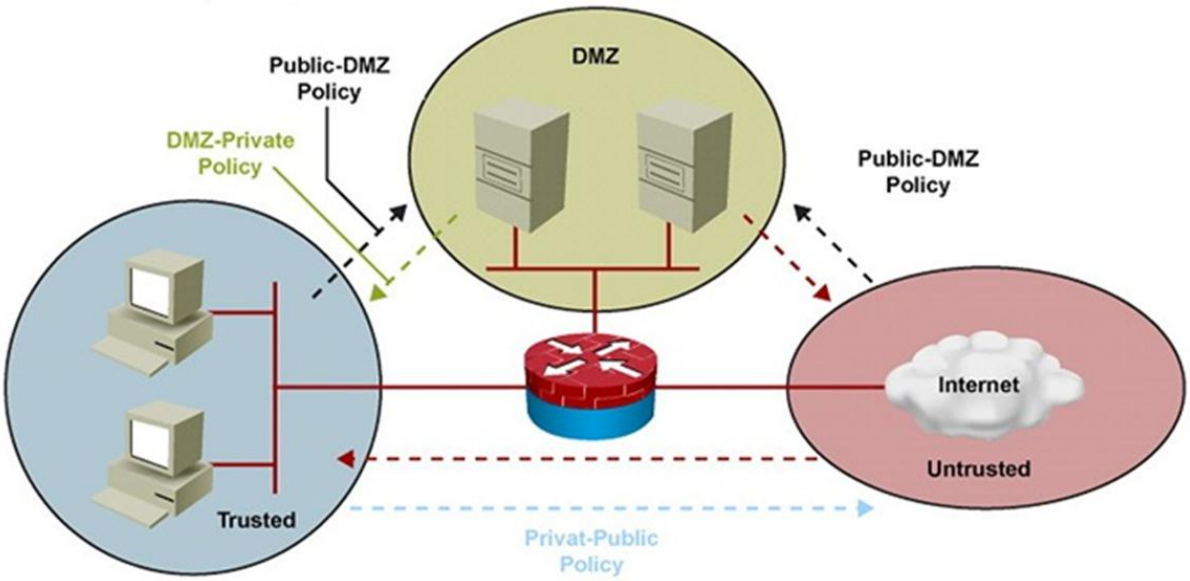
Per-policy parameter

Transparent firewall

Virtual Routing and Forwarding (VRF)-aware firewall

**ZPF Example Topology**

Zone-Based Policy firewall application with private, public and DMZ zones controlled by multiple policies.




**ZPF Configuration Example Process**

1- The following steps describe an example process for configuring a simple ZPF between the private and public zones.

2- Create an inspect class map called MY-CLASS for all TCP-based traffic that matches an ACL (Telnet, SMTP, FTP, and HTTP).

3- Create a policy map called MY-POLICY which defines the action to perform on the traffic matching the class map MY-CLASS. In this example, the action is to inspect the traffic to create dynamic inspection objects.

4- Define the two security zones, in this case PRIVATE and PUBLIC.

5- Create a corresponding zone pair called PRIV-PUB to represent the direction of the traffic to which the policy will be applied.

6- Apply the policy MY-POLICY to the zone pair to control the traffic.

7- Zone pair PRIV-PUB states that all traffic sourced from zone PRIVATE and destined for zone PUBLIC will be processed according to policy map MY-POLICY.

8- Assign each interfaces to an appropriate zone. Interface Fa0/0 is in zone PRIVATE, and interface Fa0/1 is in zone PUBLIC.

9- Define the ACL to be used by class map MY-CLASS.

```
! Define inspect class-maps:
class-map type inspect match-any TCP
 match protocol tcp
class-map type inpsect match-all MY-CLASS
 match access-group 102
 match class-map TCP
! Define inspect policy-map:
policy-map type inspect MY-POLICY
 class type inspect MY-CLASS
 inspect
! Define zones:
zone security PRIVATE
zone security PUBLIC
! Establish zone pair, apply policy:
zone-pair security PRIV-PUB source PRIVATE destination PUBLIC
 service-policy type inspect MY-POLICY
! Assign interfaces to zones:
interface FastEthernet0/0
 zone-member security PRIVATE
interface FastEthernet0/1
 zone-member security PUBLIC
! Define ACL
access-list 102 permit tcp any any eq telnet
access-list 102 permit tcp any any eq smtp
access-list 102 permit tcp any any eq ftp
access-list 102 permit tcp any any eq www
```

Other Methods of Securing the Data Plane:

Unauthorized or unwanted traffic can be blocked by implementing traffic-filtering and other security features such as:

- Standalone Access Control Lists (ACLs)

- VLAN access maps (on LAN switches)

- Cisco IOS firewall (SPI or ZPF)

- Intrusion Prevention System (IPS)

- Unicast Reverse Path Forwarding (uRPF).

- IP Security (IPsec)

- IEEE 802.1X

- Network Admission Control (NAC)

`show zone security:`
Displays information for all zones configured and corresponding member interfaces. Verifies zone configuration and assignment.

`show zone-pair security`
Provides information about how zones are paired including Zone-pair direction (with respect to the traffic flow) and policy applied to zone-pair traffic.

`show policy-map type inspect`
Displays relevant information for the policy including what traffic is matched to which class and what action is applied to each class of traffic. Also displays the dynamically created session objects.

## Branch Office & Remote Worker Connectivity Issues

### Site-to-site VPN connectivity

Misconfigured parameters causing mismatches on the VPN-termination routers.

Overlapping IP subnets on the opposite sides of the tunnel which can require the use of NAT

### Remote-access VPNs

Host issues related to client configuration or antivirus software.

User authentication and authorization is also a critical function

### GRE tunnels

Misconfiguring tunnel source and destination can cause routing issues preventing the tunnel from forming.

### Branch Office Connectivity Issues with GRE:

GRE tunnels are typically used to transport routing protocols across IPsec VPNs.

Maximum transmission unit (MTU) and fragmentation are a common issue.

Problems related to GRE tunnel establishment are usually due to configurations of tunnel sources and tunnel destinations, along with improper routing of loopbacks.

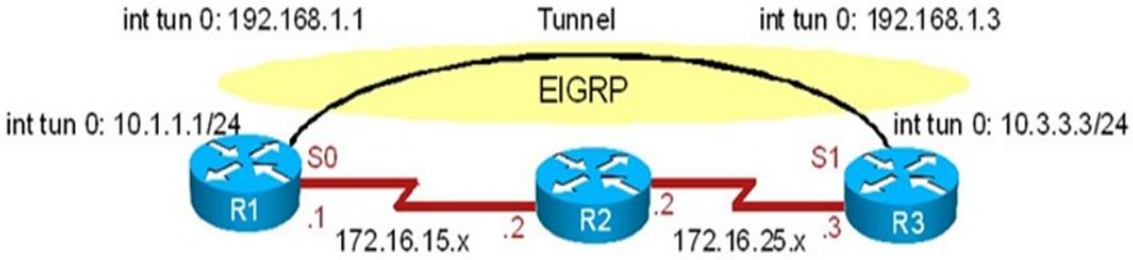
Firewalls and traffic filters may block the IPsec traffic that carries the GRE tunnels.

Multiple GRE point-to-point tunnels can saturate the physical link with routing information if the bandwidth is not adequately provisioned or configured on the tunnel interface.

The point-to-point nature of traditional GRE tunnels makes a full-mesh solution a challenge because all routers have to terminate a high number of tunnels.

Misconfiguration of routing over GRE tunnels can lead to recursive routing. In the example shown in the figure, the GRE tunnel is terminated at the loopback interfaces of the routers at each end. Those loopback interfaces are also injected into EIGRP, and they are advertised across the tunnel to the other side, from R3 to R1 and vice versa. The routing tables will show that the best path to the loopbacks, the source of the tunnel, is the tunnel itself. This causes the inconsistent routing that leads to the recursive routing problem. When the best path to the tunnel destination is through the tunnel itself, recursive routing causes the tunnel interface to flap. This might occur by just enabling a routing protocol such as EIGRP over the tunnel.

```
*Mar  1 01:54:20.075: %TUN-5-RECURDOWN: Tunnel0 temporarily disabled due to
recursive routing
```




Remote Connectivity Troubleshooting Commands

IPsec

```
show crypto ipsec sa
show crypto engine connections active
show crypto map
```

GRE

```
show interfaces tunnel
debug tunnel
```

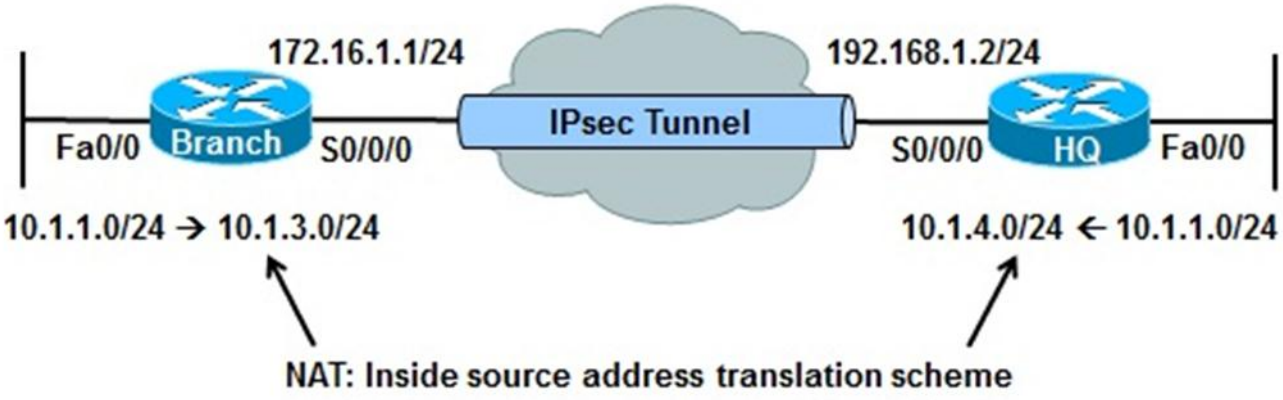
IP routing

```
show ip route
show ip protocols
debug ip routing
```

IP services

```
show ip dhcp pool
show ip dhcp bindings
show ip nat statistics
show ip nat translations
show standby
show standby brief
```

**BO/RW TSHOOT Example 1: Address Translation Error**




The Branch router is using an IPsec tunnel to provide connectivity to headquarters for its LAN users. This deployment has been working for a while, but a recent change in NAT configuration has caused the tunnel to go down, not get reestablished, and VPN connectivity to fail. This is the only branch experiencing the problem. Regular Internet access, however, has been restored, and users are able to connect to websites normally.

```
BRANCH# sh ip nat statistics
Total active translations: 1 (1 static, 0 dynamic, 0 extended)
Outside interfaces:
Serial0/0/0
Inside interfaces:
FastEthernet0/0
Hits: 0 Misses: 0
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 150 pool PUBLIC refcount 0
pool PUBLIC: netmask 255.255.255.0
start 172.16.1.100 end 172.16.1.200
type generic, total addresses 101, allocated 0 (0%), misses 0
[Id: 2] access-list VPN pool VPN_NAT refcount 0
start 10.1.10.10 end 10.1.10.200
type generic, total addresses 191, allocated 0 (0%), misses 0
Queued Packets: 0
```

The output shows that the VPN traffic is exempted from "public" translation because it remains private as it goes through the tunnel. Based on the network topology diagram, the subnets on the opposite sides of the VPN are both using address 10.1.1.0/24 and are overlapping. NAT is needed to translate VPN traffic into something other than 10.1.1.0/24 on both sides. Traffic matching the VPN access list is being statically translated into an address from the range 10.1.10.10 to 10.1.10.200. Traffic from Branch to HQ (destination subnet 10.1.4.0/24), should have its source address translate to an address from the 10.1.3.0/24 subnet. The traffic leaving the headquarters network should have its source address translated to an address from the 10.1.4.0/24 subnet.

The translation done for the VPN traffic at the branch office is incorrect. The source address is being translated to 10.1.10.x rather than 10.1.3.x. The VPN traffic being translated will eventually

go to the WAN interface to be tunneled through the IPsec VPN. The translated address must match the crypto access list; otherwise, it will not go through the VPN tunnel. Use the show crypto map command on the branch router to see the crypto ACL contained in the crypto map. This defines the traffic that will be accepted to the VPN tunnel.

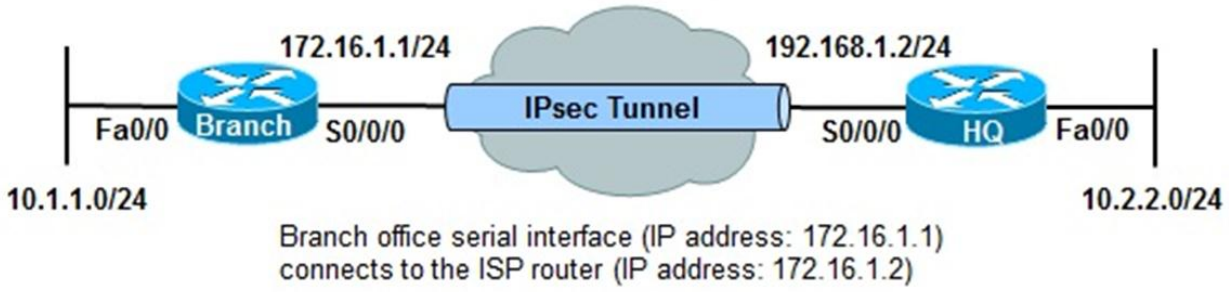
```
BRANCH# show crypto map
Crypto Map “map1” 10 ipsec-isakmp
       Peer = 192.168.1.2
       Extended IP access list 106
          access-list 106 permit ip 10.1.3.0 0.0.0.255 10.1.4.0 0.0.0.255
       Current peer: 192.168.1.2
       Security association lifetime: 4608000 kilobytes/3600 seconds
       PFS (Y/N): N
       Transform sets={
                Ts1,
       }
       Interfaces using crypto map map1:
           Serial0/0/0
```

The show crypto map command reveals that ACL 106 is used which only matches traffic with source address of 10.1.3.x and destination address of 10.1.4.x. If the source address of the traffic from the branch translates to anything other than 10.1.3.x, it will not go through the VPN tunnel. The NAT configuration is inconsistent with the crypto map (VPN) configuration.

Correct the VPN_NAT pool by removing the old definition and adding the new definition, as shown here. To test that the problem is solved, ping an address from the 10.1.4.0 pool (headquarters) with a source interface on the branch office LAN (Fa0/0) and the ping is successful.

```
BRANCH(config)# no ip nat pool VPN_NAT 10.1.10.10 10.1.10.200 netmask
255.255.255.0
BRANCH(config)#
BRANCH(config)# ip nat pool VPN_NAT 10.1.3.10 10.1.3.200 netmask 255.255.255.0
BRANCH(config)# end

BRANCH# ping 10.1.4.1 source fa0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.4.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max - 56/57/60 ms
```

**BO/RW TSHOOT Example 2: Crypto Map ACL Error**




The Branch router is using an IPsec tunnel to provide connectivity to headquarters for its LAN users. This time there is no subnet overlapping between the branch and headquarters networks. The VPN connection is down, but the Internet connection is working well. There have not been any recent documented configuration changes. This Branch router is providing DHCP services to LAN hosts.

Start at the Branch router and use a bottom-up approach for each phase or step along the path. Use the show ip interfaces brief command to check the Layer 1 and Layer 2 status of the Branch router's interfaces. As shown in the example, both the LAN and WAN interfaces are up.

```
BRANCH# sh ip int brief
Interface          IP-Address    OK?  Method   Status                  Protocol
FastEthernet0/0    10.1.1.1      YES  manual   up                      up
FastEthernet0/1    unassigned    YES  unset    administratively down  down
Serial0/0/0        172.16.1.1    YES  manual   up                      up
NVIO               unassigned    NO   unset    up                      up
```

Check whether the Branch router is providing IP address and related parameters through DHCP. The show ip dhcp pool command on the Branch router confirms that the address space 10.1.1.0/24 is being served to hosts through DHCP.

```
BRANCH# show ip dhcp pool

Pool LAN :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 254
 Leased addresses               : 0
 Pending event                  : none
 1 subnet is currently in the pool :
Current index         IP address range                 Leased addresses
10.1.1.1              10.1.1.1       - 10.1.1.254      0
```

Check to see if there is a routing problem using the show ip route command. The output show what is expected for a small branch office: a static default pointing to a next hop on the WAN interface.

```
BRANCH# show ip route
<output omitted>

Gateway of last resort is 172.16.1.2 to network 0.0.0.0

     172.16.0.0 255.255.255.0 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Serial0/0/0
     10.0.0.0 255.255.255.0 is subnetted, 3 subnets
C       10.1.3.0 is directly connected, Loopback0
C       10.1.1.0 is directly connected, FastEthernet0/0
C       10.251.1.0 is directly connected, Loopback1
S*   0.0.0.0 0.0.0.0 [1/0] via 172.16.1.2
```

Next, check NAT with the show ip nat statistics command. The output reveals that traffic matching ACL 107 will be translated.

```
BRANCH# show ip nat statistics
Total active translations: 1 (1 static, 0 dynamic, 0 extended)
Outside interfaces:
Serial0/0/0
Inside interfaces:
FastEthernet0/0
Hits: 60 Misses: 0
CEF Translated packets: 10, CEF Punted packets: 30
Expired translations: 7
Dynamic mappings:
-- Inside Source
[Id: 3] access-list 107 pool PUBLIC refcount 0
pool PUBLIC: netmask 255.255.255.0
start 172.16.1.100 end 172.16.1.200
type generic, total addresses 101, allocated 0 (0%), misses 0
```

Display ACL 107 and the content looks correct because it denies traffic going from branch to headquarters. That means the traffic going from branch to headquarters will not be subjected to NAT.

```
BRANCH# show access-list 107
Extended IP access list 107
   10 deny ip 10.1.1.0 0.0.0.255 10.2.2.0 0.0.0.255
   20 permit ip 10.1.1.0 0.0.0.255 any
```

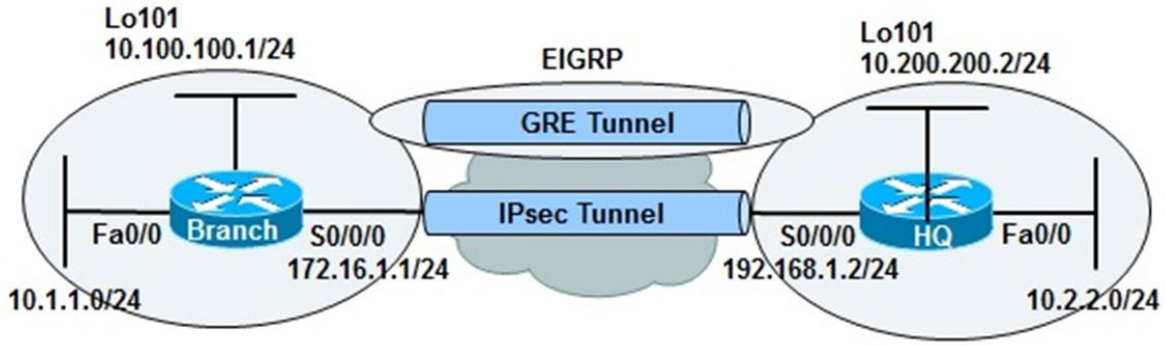
Check the VPN configuration using the show crypto map on the Branch router. ACL 106 used in the crypto map states that only the traffic with source address 10.1.3.x and destination address 10.2.2.y will go through the VPN tunnel. That is incorrect because the traffic from the branch going to the headquarters (which is not subject to NAT) will have source address of 10.1.1.x, that is provided by the DHCP server.

```
BRANCH# show crypto map
Crypto Map "map1" 10 ipsec-isakmp
        Peer = 192.168.1.2
        Extended IP access list 106
            access-list 106 permit ip 10.1.3.0 0.0.0.255 10.2.2.0 0.0.0.255
        Current peer: 192.168.1.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                ts1,
        }
        Interfaces using crypto map map1:
                Serial0/0/0
```

The source IP addresses of the packets from the branch office are not matching the crypto ACL. Change the crypto ACL 106 on Branch to permit traffic sourced from network 10.1.1.0/24 destined for network 10.2.2.0/24 to be encrypted by the tunnel. Use the ping command to verify connectivity. The ping from branch to headquarters is successful.

```
BRANCH# conf t
Enter configuration commands, one per line. End with CNTL/Z
BRANCH(config)# no access-list 106
BRANCH(config)#
BRANCH(config)# access-list 106 permit ip 10.1.1.0 0.0.0.255 10.2.2.0 0.0.0.255

BRANCH# ping 10.2.2.1 source f0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max - 88/89/92 ms
```

**BO/RW TSHOOT Example 3: GRE Config. Error**




EIGRP is being routed across an IPsec VPN tunnel, using GRE. The GRE tunnel is sourced at the loopback interfaces on each router. EIGRP is used to advertise internal networks in the 10.0.0.0 address space, for branch-to-headquarters connectivity. The problem is that traffic is not reaching the headquarters network, which hosts multiple mission-critical servers. The support team does not have many details, just that connectivity is lost.

At the Headquarters router check the status of the VPN tunnel and look for the IP address of the Branch router as a destination using the show crypto isakmp sa command. The status of the tunnel to branch at 172.16.1.1 is ACTIVE. The same command at the Branch router shows an ACTIVE status, too.

```
HQ# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id  slot  status
172.16.1.1      192.168.1.2     QM_IDLE           1002     0  ACTIVE

BRANCH# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id  slot  status
192.168.1.2     172.16.1.1      QM_IDLE           1001     0  ACTIVE
```

The VPN tunnel is reported as active from both ends. Troubleshooting from the bottom up, determine whether the headquarters destinations can be found in the Branch router's routing table. Use the show ip route command and search for network 10.2.2.0/24. This subnet is not present.

```
BRANCH# show ip route 10.2.2.0
% Subnet not in table
BRANCH#
```

Routing (advertisement) is supposed to happen over GRE across the VPN tunnel. Examine the GRE (tunnel0) using the show interfaces tunnel 0 command. The results show that the tunnel is up, but line protocol is down. The tunnel source at BRANCH is 10.100.100.1 (loopback101), and the tunnel destination is 10.200.200.22.

```
BRANCH# show interfaces tunnel 0
Tunnel0 is up, line protocol is down
  Hardware is Tunnel
  Internet address is 10.1.3.2 255.255.255.0
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
    Reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.100.100.1 (Loopback101), destination 10.200.200.22
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Fast tunneling enabled
<output omitted>
```

Check the Headquarters router and see whether address 10.200.200.22 is a valid destination for this tunnel. The show interfaces tunnel 0 command on the HQ router indicates the tunnel source at HQ is loopback101 with the IP address 10.200.200.2, not 10.200.200.22. It looks like a typing error has happened at the Branch router Notice that the tunnel interface at HQ is administratively down and that needs to be fixed, too.

```
HQ# show interfaces tunnel 0
Tunnel0 is administratively down, line protocol is down
  Hardware is Tunnel
  Internet address is 10.1.3.1 255.255.255.0
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
    Reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.200.200.2 (Loopback101), destination 10.100.100.1
  Tunnel protocol/transport GRE/IP
<output omitted>
```

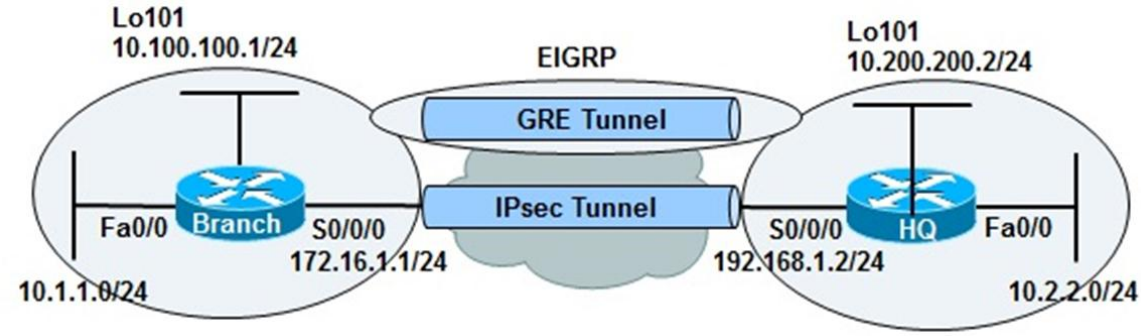
Return to the Branch router to fix the tunnel destination address error. First enter the debug ip routing command to see the EIGRP routes appear in routing table as a result of repairing the tunnel. In interface configuration mode for the tunnel0 interface, remove the incorrect tunnel destination address, and enter the correct tunnel destination address (10.200.200.2).

```
BRANCH# debug ip routing
IP routing debugging is on
BRANCH#

BRANCH# conf t
Enter configuration commands, one per line. End with CNTL/Z
BRANCH(config)# int tunnel0
BRANCH(config-if)# no tunnel destination 10.200.200.22
BRANCH(config-if)# tunnel destination 10.200.200.2
BRANCH(config-if)# end
```

Debug messages indicate the EIGRP neighbor session is established. The routing table is populated across the tunnel. Confirm end-to-end connectivity with an extended ping from the Branch router using its Fa0/0 interface as the source, to the address 10.2.2.1 at headquarters. The ping is successful.

```
BRANCH#
%SYS-5-CONFIG_I: Configured console by console
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.3.1 (Tunnel0) is up: new
adjacency
BRANCH#
%LINK-3-UPDOWN: Interface Tunnel0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up

BRANCH# ping 10.2.2.1 source f0/0
Type escape sequence to abort.
Sending 5 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 88/91/92 ms
BRANCH#
```

**BO/RW TSHOOT Example 4: Recursive Routing Issue**




The IPsec tunnel was established and tested, and it was carrying user traffic with no problem. Suddenly the tunnel interface went down and EIGRP was no longer able to advertise routes. Level 1 operators tried resetting the interfaces, but that did not help. Tunnels get established and then go down after a few seconds every time.

GRE is being used to carry EIGRP advertisements across the VPN. Use the show ip protocols command to verify the EIGRP configuration. The output is shown here and looks correct.

A display of the HQ routing table using the show ip route command includes three paths to the tunnel0 destination (10.100.100.1): The gateway of last resort (0.0.0.0/0) through 192.168.1.1 (the ISP's IP address at HQ – not shown in the topology). The one is using the EIGRP route 10.100.100.0/24 through the tunnel0 interface. The one is using the static route entered to 10.100.100.1/32 through 172.16.1.1. This is the most specific one and will be used to reach the tunnel end.

```
HQ# show ip route
<output omitted>
Gateway of last resort is 192.168.1.1 to network 0.0.0.0

     10.0.0.0 255.0.0.0 is variably subnetted, 8 subnets, 2 masks
C       10.1.3.0 255.255.255.0 is directly connected, Tunnel0
C       10.200.200.0 255.255.255.0 is directly connected, Loopback101
D       10.100.100.0 255.255.255.0
           [90/297372416] via 10.1.3.2, 00:00:07, Tunnel0
C       10.2.2.0 255.255.255.0 is directly connected, FastEthernet0/0
D       10.1.1.0 255.255.255.0 [90/297372416] via 10.1.3.2, 00:00:07, Tunnel0
S       10.100.100.1 255.255.255.255 [1/0] via 172.16.1.1
C     192.168.1.0 255.255.255.0 is directly connected, serial0/0/0
S*   0.0.0.0 0.0.0.0 [1/0] via 192.168.1.1



BRANCH# show ip protocols
Routing Protocol is “eigrp 1”
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1 =1, K2 = 0, K3 = 1, K4 = 0, K5 = 0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
Routing Information Sources:
  Gateway Distance Last Update
  (this router) 90 00:38:11
Distance: internal 90 external 170
```

Use the show interfaces tunnel command to determine the status of the tunnel on Branch. Interface Tunnel 0’s line protocol is down The source and destination of the tunnel, based on the network diagram, are correct. No tunnel configuration error is apparent. The same command on the HQ router shows correct configuration, but the line protocol is down there, too.

```
BRANCH# show interface tunnel 0
Tunnel0 is up, line protocol is down
  Hardware is Tunnel
  Internet address is 10.1.3.2 255.255.255.0
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
    Reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.100.100.1 (Loopback101), destination 10.200.200.2
  Tunnel protocol/transport GRE/IP
```

Replicate the problem by shutting down the interfaces on HQ and bringing them back up. This will initiate establishment of the tunnel. An informational message on a new adjacency with the neighbor Branch across tunnel0 is reported. After a few seconds another message displays:

“Tunnel0 temporarily disabled due to recursive routing.” The line protocol on interface tunnel0 changes state to down, and so does the neighbor.

```
HQ(config)# int tunnel0
HQ(config-if)# shutdown
HQ(config-if)# no shutdown
HQ(config-if)# end
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.3.2 (Tunnel0) is up: new
adjacency
HQ#
%TUN-5-RECURDOWN: Tunnel0 temporarily disabled due to recursive routing
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to down
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.3.2 (Tunnel0) is down:
interface
down
```

Configure a path to the tunnel destination that is better than the EIGRP path through the tunnel itself using a static route. BRANCH address 172.16.1.1 (assumed to be a public address) is considered to be the address the ISP has assigned to the BRANCH router, and the HQ address 192.168.1.2 (assumed to be a public address) is considered to be the address that HQ’s ISP has assigned to the HQ router. The tunnel interface goes up and neighbor adjacency is established.

```
HQ(config)# interface tunnel0
HQ(config-if)# shutdown
HQ(config-if)# exit
HQ(config)# ip route 10.100.100.1 255.255.255.255 172.16.1.1
HQ(config)# interface tunnel0
HQ(config-if)# no shutdown
HQ
%LINK-3-UPDOWN: Interface Tunnel0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up
HQ#
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.3.2 (Tunnel0) is up: new
Adjacency
```

**BO/RW TSHOOT Example 5: ACL Denies IPsec**


A security auditor recently performed a security assessment and recommended a few improvements to the network policy. After the change, IPsec tunnels do not work and never get established. VPN connectivity is critical for branch services. All configurations have been reverted to their pre-audit state, except for the Branch router. Cisco IOS firewall services were installed in some important routers of the network.

Use the show ip interfaces command to determine if ACLs are applied to any interface. The output shows that interface s0/0/0 is up/up and an ACL called FIREWALL-INBOUND is applied in the inbound direction. This interface is the one that terminates the IPsec tunnel.

```
BRANCH# show ip interface s0/0/0
Serial0/0/0 is up, line protocol is up
  Internet address is 172.16.1.1 255.255.255.0
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound access list is FIREWALL-INBOUND
  Proxy ARP
<output omitted>
```

The ACL is allowing routing protocols and management protocols such as SSH. The ACL is missing statements that permit IPsec protocols and ISAKMP. IPsec requires ESP/AH (protocols 50/51) and ISAKMP (UDP Port 500) to be allowed by access lists. The ACL is blocking those ports.

```
BRANCH# show access-list FIREWALL-INBOUND
Extended IP access list FIREWALL-INBOUND
   10 permit tcp any 192.168.250.16 0.0.0.15 established
   20 permit tcp any host 192.168.250.16 eq www
   30 permit tcp any any eq 22
   40 permit tcp any any eq telnet
   50 permit tcp any host 192.168.250.16 eq ftp
   60 permit icmp any any
   70 permit eigrp any any (120 matches)
```

Add the required lines to the ACL, and also add a remark indicating why you are making this change using the access-list remark command. Three IPsec protocols should be allowed:

- ESP

- AHP

- ISAKMP

Verify the solution by successfully pinging the HQ router (loopback interface), which is learned through the tunnel, from the Branch router.

```
BRANCH# configure terminal
Enter configuration commands, one per line. End with CNTL/Z
BRANCH(config)# ip access-list extended FIREWALL-INBOUND
BRANCH(config-ext-nacl)# remark --additions for IPsec ---
BRANCH(config-ext-nacl)# permit udp any any eq 500
BRANCH(config-ext-nacl)# permit esp any any
BRANCH(config-ext-nacl)# permit ahp any any
BRANCH(config-ext-nacl)# end
```